AÑO III - NUM. 74

MICROHOBBY

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES SINCLAIR

SEMANAL

AÑO III - N.º 74

135 PTS.
Canarias 140 ptas.



NUEVO

"SWEEVO'S WORLD"

La increíble aventura de un torpe robot

EXPANSION

ART STUDIO:

EL MEJOR DISEÑADOR GRÁFICO CREADO PARA SPECTRUM

UTILIDADES

UNA RUTINA QUE CHEQUEA LA MEMORIA

¡Y TE INDICA LOS FALLOS!

TOP SECRET

PROTECCIONES A NIVEL BASIC

PROFESOR PARTICULAR

RESOLUCIÓN DE ECUACIONES

HOBBY PRESS


GREEN BERET
imagine
Konami
DISTIBUIDOR EXCLUSIVO PARA ESPAÑA ERBE SOFTWARE
C/. STA. ENGRACIA, 17. 28010 MADRID. TFNO.: (91) 447 34 10
DELEGACION BARCELONA. AVDA. MISTRAL, N.º 10 - TFNO.: (93) 432 07 31

Director Editorial
José I. Gómez-Centurión
Director Ejecutivo
Domingo Gómez
Asesor Editorial
Gabriel Nieto
Redactora Jefe
Africa Pérez Tolosa
Diseño
Rosa María Capitel
Redacción
Amalio Gómez, Pedro Pérez,
Jesús Alonso
Secretaría Redacción
Carmen Santamaría
Colaboradores
Primitivo de Francisco, Rafael Prades,
Miguel Sepúlveda, Sergio Martínez
y J. M. Lazo
Corresponsal en Londres
Alan Heap
Fotografía
Carlos Candel
Portada
José María Ponce
Dibujos
Teo Mójica, F. L. Frontán,
J. M. López Moreno,
J. Igual, J. A. Calvo,
Lóriga, J. Olivares

Edita
HOBBY PRESS, S. A.
Presidente
María Andrino
Consejero Delegado
José I. Gómez-Centurión
Jefe de Producción
Carlos Peropadre
Jefe de Publicidad
Concha Gutiérrez
Publicidad Barcelona
José Galán Cortés
Tels.: 303 10 22 - 313 71 76
Secretaría de Dirección
Marisa Cogorro
Suscripciones
M.ª Rosa González
M.ª del Mar Calzada
Redacción, Administración y Publicidad
La Granja, 39
Polígono Industrial de Alcobendas
Tel.: 654 32 11
Telex: 49480 HOPR
Dto. Circulación
Paulino Blanco
Distribución
Coedis, S. A. Valencia, 245
Barcelona
Imprime
Rotedic, S. A. Ctra. de Irún,
km 12,450 (MADRID)
Fotocomposición
Novocomp, S.A.
Nicolás Morales, 38-40
Fotomecánica
Grof
Ezequiel Solana, 16
Depósito Legal
M-36.598-1984

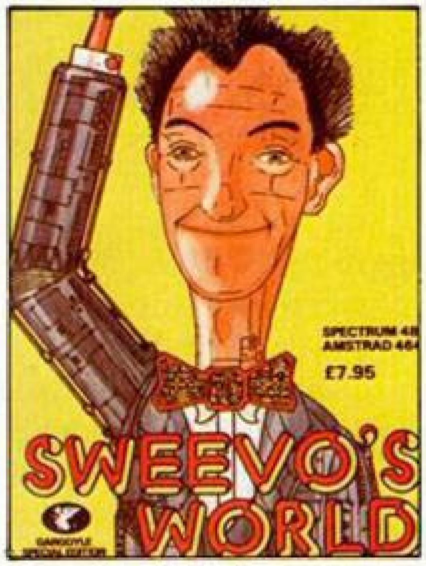
Representante para Argentina, Chile, Uruguay y Paraguay, Cía. Americana de Ediciones, S.R.L. Sud América 1.532. Tel.: 21 24 64. 1209 BUENOS AIRES (Argentina)

Solicitado control
OJD

MICROHOBBY ESTA SEMANA

AÑO III. N.º 74. 15 al 21 de abril de 1986.
135 ptas. Canarias, Ceuta y Melilla: 130 ptas.
Sobretasa aérea para Canarias: 10 ptas.





«Sweev'os World», las andanzas del robot Sweevo. (Pág. 12)

PREMIADOS DEL CONCURSO «OLE, TORO»

1. José González Villarreal. Madrid.
2. Gonzalo Sánchez de Pablo. Sevilla.
3. H. Sharaf. Madrid.
4. Germán Fabregat Llueca. Valencia.
5. Rafael de la Piñera Santoro. Madrid.
6. Miguel Angel Navarro Molina. Albacete.
7. Fernando María Manero Vargas. Colmenar Viejo. (Madrid).
8. Francisco Alejandre Cabrerizo. Guadalajara.
9. Ignacio Lacadena García Gallo. Madrid.
10. Luis Algorta Bellón. Bilbao.

SINCLAIR-AMSTRAD, una nueva relación comercial

La controvertida situación económica que Sinclair Research venía arrastrando en los últimos meses, ha tocado fondo de una manera definitiva. El resultado: la fragmentación en varias compañías diferentes y la compra, por parte de Amstrad, de todas las actividades relativas a la comercialización y distribución de sus ordenadores personales.

Esta radical reestructuración, por otra parte esperada, ha dado como resultado la creación de varias áreas de trabajos dentro de Sinclair, pero perfectamente diferenciadas unas de otras. Estas han sido las siguientes: desarrollo e investigación en el campo de los micro-ordenadores, tecnologías de alta escala de integración en semiconductores y desarrollo de productos innovadores en el campo de las telecomunicaciones.

Esta cesión de derechos a Amstrad se refiere, únicamente, a la fabricación y comercialización de sus micro-ordenadores, pero no a la propia realización y diseño de los mismos. De esta forma, Sinclair continuará desarrollando nuevos modelos, entre los que se incluyen los proyectos actuales del ordenador portátil **Pandora**, basado en un sistema de pantalla plana integrada, el megacomputer **Enigma** y otras posibles creaciones a más largo plazo.

Según palabras del propio sir Clive Sinclair, «**estamos estructurando una nueva organización con el fin de imprimir una mayor flexibilidad y potencia a nuestros productos e investigaciones. Cada negocio satélite será capaz de atraer por sí mismo a una buena cantidad de inversores e investigadores. Del mismo modo tenemos pensado continuar creando equipos de trabajo en cualquier área tecnológica...**».

Al hacerse cargo Amstrad, a través de su director gerente Allan Sugar, de la comercialización de los ordenadores Sinclair, lo que ha supuesto un desembolso de cinco millones de libras (unos mil millones de pesetas), ha quedado asegurado en cierta manera la continuidad de la difusión de dichos productos en el mercado, en contra de lo que en un principio se podría pensar de que esta drástica medida supondría la progresiva e imparable desaparición definitiva de la empresa.

Este suceso ha sido acogido en Gran Bretaña con verdadero optimismo, y tanto los ambientes comerciales como profesionales han visto con muy buenos ojos el hecho de que un excelente comerciante y hombre de negocios como Allan Sugar se haya hecho cargo de los derechos mundiales de los productos de Sinclair.

Confiamos firmemente en que esta medida contribuirá a mejorar aún más el ya mítico nombre de Sinclair y que esta acción garantizará una fructífera colaboración entre ambas firmas, que verán finalizada de una vez por todas su histórica competencia.

MICROHITS

1. **STARQUAKE** (Bouble Bus)
2. **FAIRLIGHT** (The Edge)
3. **YIE AR KUNG FU** (Imagine)
4. **COMANDO** (Elite)
5. **MARSPORT** (Gargoyle Games)
6. **ROBIN OF THE WOOD** (Odin)
7. **ELITE** (Firebird)
8. **SIR FRED** (Made in Spain)
9. **ROLLER COASTER** (Elite)
10. **CAMELOT WARRIORS** (Dinamic)
11. **MICKIE** (Imagine)
12. **SABOTEUR** (Durell)
13. **GUNFRIGHT** (Ultimate)
14. **BATTLE OF THE PLANETS** (Mikro-Gen)
15. **TOMAHAWK** (Digital)
16. **SWEEVO'S WORLD** (Gargoyle Games)
17. **TRANSFORMERS** (Ocean)
18. **THREE WEEKS IN PARADISE** (Mikro-Gen)
19. **NOMAD** (Ocean)
20. **YABBA DABBA DOO** (Quicksilva)

Organizada por Indescomp en Madrid
PRIMERA FERIA AMSTRAD

Durante el transcurso del próximo mes de mayo, Indescomp, (distribuidora oficial de los productos Amstrad en España), organizará en Madrid la 1.ª Feria Amstrad.

El principal objetivo de dicha feria será el de dar a conocer al público en general la amplia gama de modelos que Amstrad posee actualmente en el mercado, así como cualquier otro tipo de productos relacionados directamente con ellos, tal y como ha ocurrido recientemente en otras ciudades como París y Londres.

Otra de las intenciones principales de esta 1.ª Feria Amstrad, será la de servir de punto de encuentro entre las compañías de otros países que tengan interés en presentar y comercializar sus periféricos, libros, software, etc...

Un acontecimiento muy importante que esperamos que sirva de ejemplo para que la distribuidora española de Sinclair realice en un futuro próximo una feria similar.

GARGOYLE GAMES REALIZARA PAQUETES DE UTILIDADES

Gargoyle Games, compañía de software sobradamente conocida en nuestro país, ha anunciado en estos días que, movida por la intención de lanzar al mercado una mayor cantidad de productos originales y de calidad, tiene pensado presentar en la próxima PCW Show, feria de informática que se realiza anualmente en Londres, el lanzamiento de un nuevo sello de programas.

Este sello estará formado por un equipo de programadores cuya dedicación se centrará principalmente en la creación de paquetes de utilidades para diferentes ordenadores, aunque por el momento comenzarán con Amstrad y con el IBM PC (1).

En cuanto a lo referente a los programas de entretenimiento, Gargoyle Games tiene pensado el lanzamiento de «Heavy on the Magick», una aventura animada en la que se mezclarán el diálogo con el ordenador y una serie de complejas secuencias animadas.



«Heavy on the Magick» vendrá acompañado de una fuerte campaña de lanzamiento y aparecerá en su versión para Spectrum durante el presente mes de abril, mientras que los usuarios de Amstrad y Commodore tendrán que esperar hasta mayo para poder disfrutar de este, «a priori», interesante juego.

(1) Según palabras del mismo Greg Follis, director de la compañía, «Si Gargoyle es el Rolls Royce de los juegos para ordenadores personales, el nuevo sello se convertirá en un auténtico Jaguar.»

AQUI LONDRES



Popeye, programa que fue confeccionado por DK'Tronics, pero que nunca fue publicado en Inglaterra, ha vuelto a aparecer ahora por cortesía de McMillan Ltd. De momento, el juego se puede conseguir para el Spectrum 48 y 128 K y las versiones para C64 y Amstrad aparecerán a lo largo de este año.

Spectravídeo, la compañía famosa por su joystick «Quickshot», va a lanzar pronto un nuevo producto: **un brazo robot.** El robot, que está orientado principalmente hacia la gente joven, tiene una gran variedad de movimientos y puede hacer operaciones como agarrar, sujetar y recoger objetos, es controlado por ordenador y Spectravídeo tiene planeado fabricar interface para Spectrum, Commodore y Amstrad. Su precio será de 40 libras.

La próxima microferia ZX tendrá lugar, de nuevo, en el Royal Horticultural Hall de Londres, el día 10 de mayo.

Cauldron II, segunda parte del famoso Cauldron, y recientemente considerado el juego del año por una revista francesa, será lanzado por **Palace Soft**, el día 23 de mayo. El nuevo juego tiene un estilo parecido al de su predecesor (una aventura arcade) con un total de 128 pantallas. Su precio será de 9 libras y en principio se lanzará para el C64, si bien las versiones para Spectrum y Amstrad podrán conseguirse un mes después de su fecha de lanzamiento.



Ocean, ha publicado un nuevo programa para el Spectrum llamado **Bathman.** Basado en el personaje de comics del mismo nombre, está muy bien escrito con gráficos altamente detallados. Pronto lo veremos en las listas de éxitos.

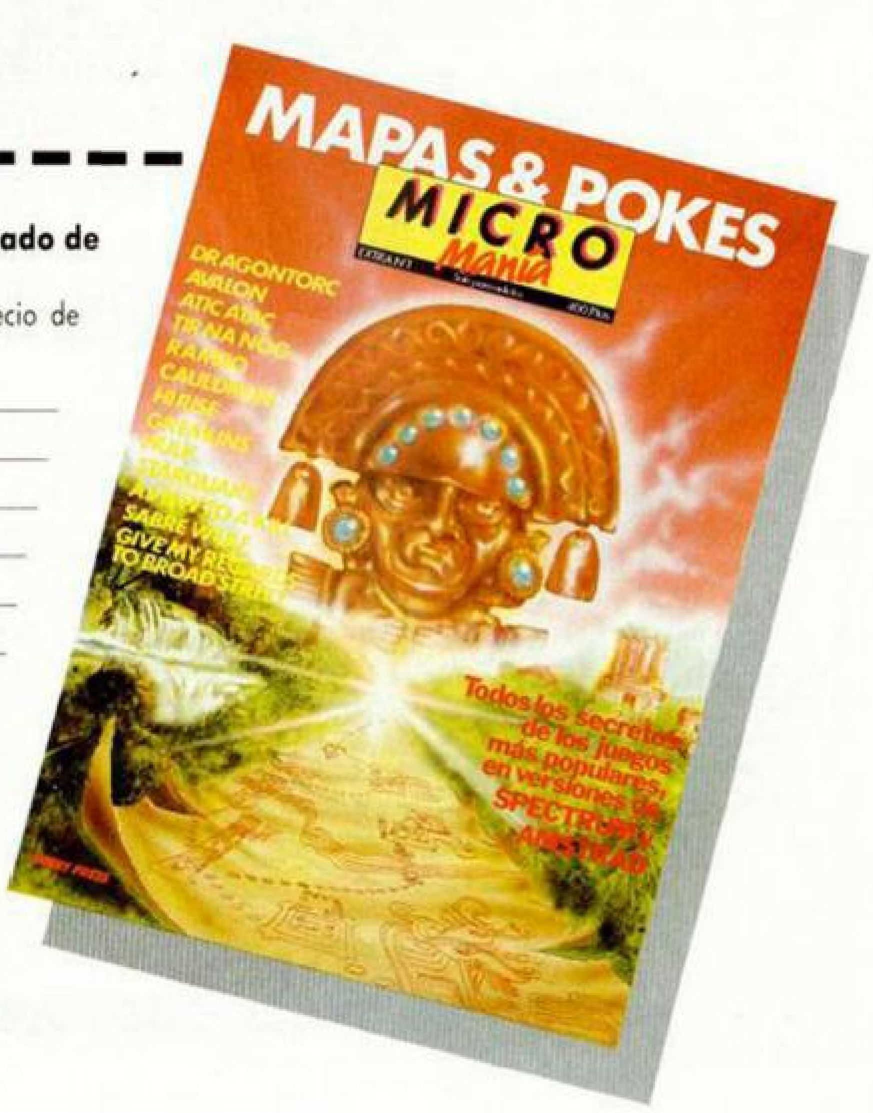
De nuestro corresponsal en Londres
ALAN HEAP

UN NUMERO ESPECIAL!!

Con todos los secretos de tus juegos preferidos

HULK
HI RISE
GIVE MY REGARDS TO BROAD STREET
GREMLIS
AVALON
RAMBO
DRAGONTORC
STARQUAKE
A VIEW TO A KILL
SABRE WULF
CAULDRON
ATIC ATAC
TIR NA NOG

MAPAS & POKES
A todo color



Recorta o copia este cupón y envíalo a Hobby Press, S.A. Apartado de Correos 232. Alcobendas (Madrid).

Deseo recibir en mi domicilio el **Especial de Micromanía n.° 1** al precio de **400 ptas.** (I.V.A. incluido).

Nombre ______________________ Edad ________

Apellidos ______________________________

Domicilio ______________________________

Localidad __________________ Provincia ________

C. Postal ________ Teléfono ________ Profesión ________

¿Eres suscriptor de **Micromanía?** ________________

Forma de pago (marco con una cruz la forma que más me interesa).

- ☐ Mediante giro postal a nombre de Hobby Press, S.A. n.° ________
- ☐ Mediante talón a nombre de Hobby Press, S.A.
- ☐ Contra reembolso (supone 75 ptas. de gastos de envío).

Fecha y firma

TRUCOS

ANTIBREAK

Este programa en Código Máquina, que nos ha enviado Juan José Rosado, os será muy útil para la protección de programas.

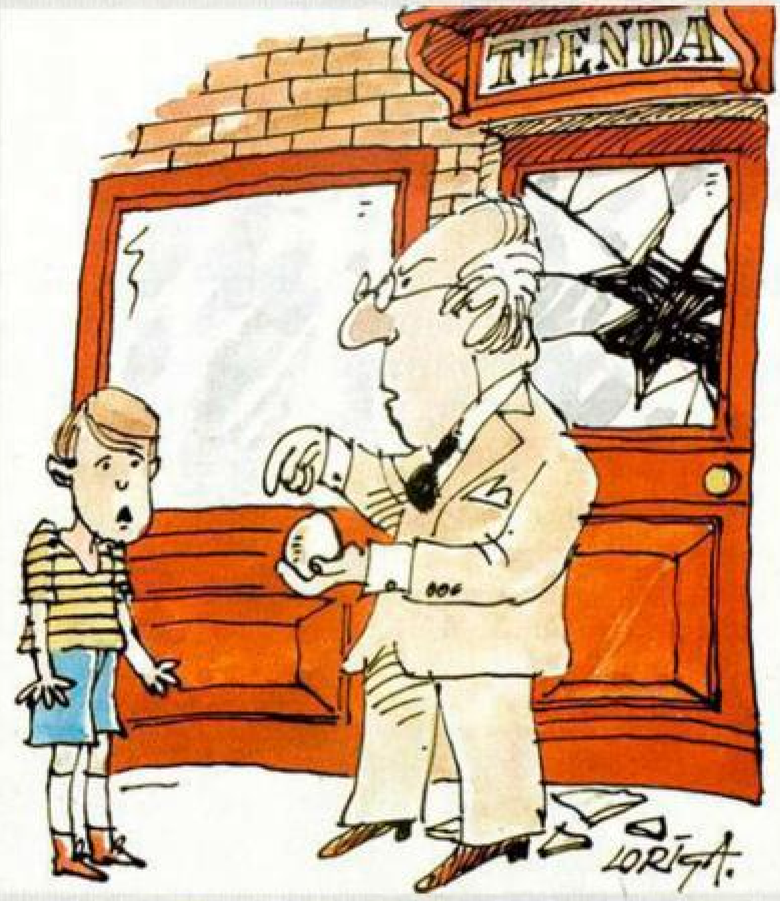
Se trata de una rutina basada en las interrupciones del Z 80. El programa lee las teclas CAPS SHIFT y SPACE varias veces por segundo, al mismo tiempo que funciona un programa Basic (o también en Código Máquina). Si dichas teclas están pulsadas, se produce un borrado total de la memoria con lo que no se podrá listar el programa. Aquí os ofrecemos el listado en Asembler, pero tened cuidado porque la rutina no es reubicable.

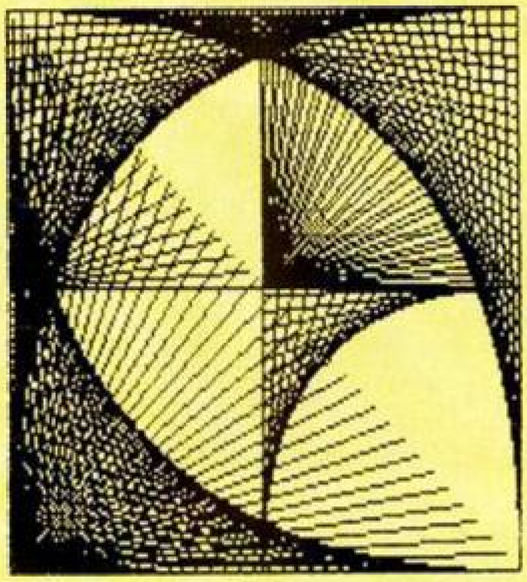
```
10        ORG  64255
20        DEFW 64280
30        NOP
40        NOP
50        NOP
60        DI
70        NOP
80        NOP
90        NOP
100       NOP
110       NOP
120       NOP
130       LD   A,250
140       LD   I,A
150       IM   2
160       EI
170       RET
180       DI
190       IM   1
200       EI
210       RET
220       DI
230       PUSH AF
240       PUSH DE
250       PUSH HL
260       PUSH BC
270       RST  56
280       DI
290       CALL UNO
300       POP  BC
310       POP  HL
320       POP  DE
330       POP  AF
340       EI
350       RET
360  UNO  CALL 654
370       DI
380       XOR  A
390       LD   A,D
400       CP   39
410       RET  NZ
420       LD   A,E
430       CP   32
440       RET  NZ
450       RST  0
```

Para los que no tienen ensamblador, aquí os ofrecemos el cargador Basic.

Para ejecutar la rutina, teclear RANDOMIZE USR 64267

```
  10 CLEAR 64253
  20 FOR F=64255 TO 64309
  30 READ A: POKE F,A
  40 NEXT F
  50 DATA 24,251,0,0,0,243,0,0,0
,0,0,0,62,250,237,71,237,94,251,
201,243,237,86,251,201,243,245,2
13,229,197,255,243,205,40,251,19
3,225,209,241,251,201,205,142,2,
243,175,122,254,39,192,123,254,3
2,192,199
```





```
  10 FOR a=0 TO 175 STEP 5
  20 PLOT 40,0: DRAW 175-a,a
  30 PLOT 40,a: DRAW 175-a,-a
  40 PLOT 40,a: DRAW a,175-a
  50 PLOT a-215,175: DRAW a,a-17
5
  60 NEXT a
  70 FOR a=0 TO 87 STEP 5
  80 PLOT 40,87: DRAW 87-a,a
  90 PLOT 127,87: DRAW 87-a,a
 100 PLOT 127,a: DRAW a,87-a
 110 PLOT 127.87: DRAW a,87-a
 120 NEXT a
```

CURVAS Y RECTAS

Este curioso efecto de curvas y rectas se lo debemos a Félix Martín y a él sólo hay que hacer una observación: si cambiamos el valor numérico del STEP (que en el programa es el 5) por otro número como el 2, el 3, el 10 u otro cualquiera, saldrá la misma figura, pero con otro dibujo. Intentarlo y veréis.

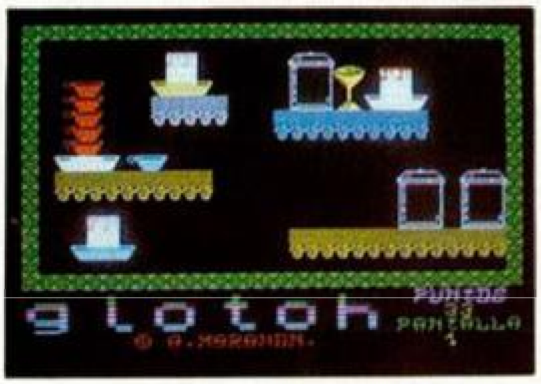
GLOTON

Adolfo MARAÑON

Spectrum 48 K

Un terrible acontecimiento ha tenido lugar en la pastelería del señor Tomás, una plaga de mosquitos glotones se ha afincado en sus pasteles.

Uno a uno, sus deliciosos pasteles se han plagado de estos desagradables insectos que prometen acabar con las existencias en poco tiempo. Ante esta situación catastrófica, Don Tomás ha reclamado nuestra ayuda para ir eliminando, con nuestro fulminante spray, cada uno de los mosquitos que ocupan las diferentes dependencias de la pastelería, formando a su paso distintos gráficos en la pantalla. Ponte manos a la obra y ¡buena suerte!







```
   2 REM © A.MARANON 1985
   3 POKE 23658,8
   5 BORDER 0: PAPER 0: INK 0: C
LS : GO SUB 6000
   7 POKE 23675,8: POKE 23676,25
4
   8 GO SUB 5000: RANDOMIZE
  10 RESTORE 11
  11 DATA 33,100,2,229,17,0,0,20
5,181,3,225,17,250,255,25,124,60
,32,240,201
  12 FOR n=23301 TO 23320: READ
q: POKE n,q: NEXT n
  40 POKE 23675,88: POKE 23676,2
55
  50 LET x=x+SGN (h-x): LET y=y+
SGN (z-y)
  55 LET j=j+2: IF j>=4 THEN LET
 j=0
  60 IF ATTR (x,y)=122 THEN LET
 h=INT (RND*16)+1: BEEP .02,30:
BEEP .03,20: LET cm=USR 23301: B
EEP .03,40: LET z=INT (RND*29)+1
: PRINT AT x,y;" ": LET x=x+SGN
(h-x): LET y=y+SGN (z-y): LET ta
r=tar+1: LET total=total-1: LET
t$=STR$ total: PRINT AT 19,27; I
NK 7;t$+" ": IF (tar=i(g)) OR (t
otal=33) THEN GO SUB 1500
  70 IF x=h THEN IF y=z THEN LET
 h=INT (RND*16)+1: LET z=INT (RN
D*29)+1
  80 PRINT AT a,b;" ": IF ATTR (
x,y)=0 THEN LET a=x: LET b=y: PR
INT AT x,y; INK 4;"K"
  90 IF INKEY$="M" THEN IF ATTR
(l,c+2)=0 THEN IF ATTR (l+1,c+2)
=0 THEN LET v=2: LET c=c+1: LET
f$="H": LET k$="GF": LET q$="LHI
H"
  95 IF INKEY$="N" THEN IF ATTR
(l,c-1)=0 THEN IF ATTR (l+1,c-1)
=0 THEN LET v=1: LET c=c-1: LET
f$="J": LET k$="BB": LET q$="CDC
E"
 100 IF INKEY$="A" THEN IF ATTR
(l-1,c)=0 THEN IF ATTR (l-1,c+1)
=0 THEN LET l=l-1
 110 IF INKEY$="" THEN IF ATTR (
l+2,c)=0 THEN IF ATTR (l+2,c+1)=
0 THEN LET l=l+1
 120 IF O<>L OR P<>C THEN PRINT
AT O,P; INK 0;"  ";AT O+1,P;"  "
: LET O=L: LET P=C
 125 PRINT AT l,c; INK 7;k$;AT l
+1,c;q$(j+1 TO j+2)
 140 IF INKEY$="Z" THEN IF ATTR
(l,c+s(v))=0 THEN PRINT AT l,c+s
(v); INK 6;f$: LET cm=USR 23301:
 PRINT AT l,c+s(v); PAPER 0; INK
 0;" ": IF l=x AND c+s(v)+w(v)=y
 THEN GO SUB 1000
 200 GO TO 50
1000 LET g=g+1: FOR n=x TO 16
1010 PRINT AT n,y; INK 0; PAPER
0;" "
1020 IF ATTR (n+1,y)<>0 THEN PRI
NT INK 2;AT n,y;"K": GO TO 1045
1030 PRINT AT n+1,y; INK 2;"K":
BEEP .02,50
1040 NEXT n
1045 LET tar=0: LET pun=pun+1
1046 LET l=1: LET c=1: LET x=13:
 LET y=29: LET a=13: LET b=29: L
ET o=1: LET p=1
1050 FOR n=1 TO 16: PRINT AT n,1
; INK 0; PAPER 0;"
                    ": NEXT n
1060 POKE 23675,8: POKE 23676,25
4: GO SUB (8000 AND pun=7)+(5246
 AND pun=6)+(5010 AND pun=2)+(50
60 AND pun=3)+(5100 AND pun=4)+(
5200 AND pun=5): POKE 23675,88:
POKE 23676,255: LET pan=pan+1: P
RINT AT 21,27; INK 7;pan;" ": RE
TURN
1500 REM fin tarta pantalla
1502 POKE 23675,8: POKE 23676,25
4
1505 LET tar=0
1510 GO SUB 8100
1515 GO TO 5
5000 REM pantallas
5001 REM pant. 1
5010 PRINT AT 13,23; INK 7; PAPE
R 0;"C[redacted]D";AT 15,22; PAPER 0;
 INK 5;"AAAAAAAAA";AT 14,22;"BBB
BBBBBB": FOR n=11 TO 12: PRINT P
APER 7; BRIGHT 1; INK 0;AT n,24;
"  ": NEXT n: PRINT AT 10,24;
 INK 2; PAPER 7; BRIGHT 1;"GAGGG
"
5020 PRINT AT 7,2; INK 5; PAPER
0;"C[redacted]D";AT 9,1; PAPER 0; INK 5
;"AAAAAAA";AT 8,1;"BBBBBBB": FOR
 n=5 TO 6: PRINT PAPER 7; BRIGHT
 1; INK 0;AT n,3;"  ": NEXT n:
PRINT AT 4,3; INK 3; PAPER 7; BR
IGHT 1;"GAG"
5030 FOR n=6 TO 10: PRINT INK 2;
 PAPER 6;AT n,11;"ECD": NEXT n:
PRINT AT 10,14; INK 6;"C[redacted]D";AT
9,14; INK 5;"ECD": PRINT AT 11,1
0; INK 5; PAPER 0;"BBBBBBBBBBB";A
T 12,10;"AAAAAAAAAAA"
5040 PRINT AT 4,19; INK 5; PAPER
 0;"BBBBBBB";AT 5,19;"AAAAAAA";A
T 3,22; INK 6;"C[redacted]D"
5050 RETURN
5055 REM pant.2
5060 PRINT AT 14,5; INK 5; PAPER
 0;"C[redacted]D";AT 16,3; PAPER 0;
 INK 5;"AAAAAAAAAAAAA";AT 15,3;"B
BBBBBBBBBBBB": PRINT PAPER 7; BRI
GHT 1; INK 0;AT 13,6;"  ":
NEXT n: PRINT AT 12,6; INK 2; PA
PER 7; BRIGHT 1;"GAGGAGG"
5075 FOR n=2 TO 7 STEP 3: PRINT
```



```
INK 6; PAPER 0;AT n,11;"HIHIHI";
AT n+1,11;"JKJKJK";AT n+2,11;"LM
LMLM": NEXT n: PRINT AT 7,17; IN
K 7;"C[redacted]D";AT 7,21; INK 5;"ECD":
 PRINT AT 8,10; INK 7; PAPER 0;"
BBBBBBBBBBBBBB";AT 9,10;"AAAAAAA
AAAAAAA"
5080 PRINT AT 7,1; INK 5; PAPER
0;"BBBBBB";AT 8,1;"AAAAAA";AT 6,
2; INK 4;"C[redacted]D";AT 5,3; INK 7; B
RIGHT 1;"[redacted]";AT 4,3; INK 2; PAPE
R 7;"GG"
5090 PRINT AT 13,20; INK 5; PAPE
R 0;"C[redacted]D";AT 15,20; INK 6;"AAAA
AA";AT 14,20;"BBBBBB"
5095 PRINT AT 5,27; INK 2; PAPER
 0;"C[redacted]D";AT 7,26; INK 6;"AAAAA"
;AT 6,26;"BBBBB";AT 4,28; INK 0;
 BRIGHT 1; PAPER 7;"  ";AT 3,28;
 INK 3;"GG"
5099 RETURN
5100 REM pant.lab.
5105 PRINT AT 16,26; INK 5; PAPE
R 0;"C[redacted]D";AT 15,27; PAPER 7; BR
IGHT 1; INK 0;"  ";AT 14,27; INK
 3;"GG"
5110 INK 1: PAPER 0: FOR n=1 TO
 2: PRINT AT n,1;"EEEEEEEEEEEE":
 NEXT n
5115 FOR n=5 TO 6: PRINT AT n,1;
"FFFFFFFF": NEXT n
5120 FOR n=7 TO 8: PRINT AT n,15
;"FFFFFFFF": NEXT n
5125 FOR n=13 TO 14: PRINT AT n,
5;"FFFFFFFF": NEXT n
5128 FOR n=9 TO 10: PRINT AT n,5
;"FFFFFFFF": NEXT n
5130 FOR n=11 TO 12: PRINT AT n,
19;"FFFFFFFF": NEXT n
5135 FOR n=15 TO 16: PRINT AT n,
15;"FFFFFFFF": NEXT n
5140 FOR n=6 TO 7: PRINT AT n,25
;"FFFFFF": NEXT n
5145 PRINT AT 1,15;"EEEEEEEEEEEE
": NEXT n
5150 FOR n=3 TO 8: PRINT AT n,11
;"FF": NEXT n
5155 FOR n=9 TO 12: PRINT AT n,1
;"FF": NEXT n
5160 FOR n=9 TO 14: PRINT AT n,1
5;"FF": NEXT n
5165 FOR n=1 TO 4: PRINT AT n,15
;"FFFF": NEXT n
5175 FOR n=1 TO 12: PRINT AT n,2
5;"FF": NEXT n
5180 FOR n=4 TO 6: PRINT AT n,21
;"FF": NEXT n
5185 FOR n=3 TO 4: PRINT AT n,25
; INK 0; PAPER 0;"  ": NEXT n
5186 INK 0: PAPER 0
5190 RETURN
5195 REM pant.4
5200 FOR n=4 TO 8: PRINT INK 2;
PAPER 8;AT n,2;"ECD": NEXT n: PR
INT AT 9,2; INK 7;"C[redacted]D";AT 9,6;
 INK 5;"ECD": PRINT AT 10,2; INK
 6; PAPER 0;"BBBBBBBBBB";AT 11,2
;"AAAAAAAAAA"
5210 PRINT AT 3,20; INK 6; PAPER
 0;"HI";AT 4,20;"JK";AT 5,20;"LM
"; INK 7; PAPER 0;AT 2,17;"NPO";
AT 3,17;"Q"; PAPER 0; INK 0;" ";
 INK 7;"R";AT 4,17;"Q"; INK 0;"
"; INK 7;"R";AT 5,17;"SUT": PRIN
T AT 6,16; INK 5; PAPER 0;"BBBBB
BBBBB";AT 7,16;"AAAAAAAAAA"; INK
 7;AT 5,22;"C[redacted]D"; BRIGHT 1; PAP
ER 7; INK 0;AT 4,23;"  "; INK 2;
AT 3,23;"GG"
5220 PRINT INK 7; PAPER 0;AT 10,
24;"NPO";AT 11,24;"Q"; INK 0;" "
; INK 7;"R";AT 12,24;"Q"; INK 0;
" "; INK 7;"R";AT 13,24;"SUT": P
RINT AT 14,17; INK 6; PAPER 0;"B
BBBBBBBBBBBBB";AT 15,17;"AAAAAAA
AAAAAAA"; INK 7;AT 10,28;"NPO";A
T 11,28;"Q"; INK 0;" "; INK 7;"R
";AT 12,28;"Q"; INK 0;" "; INK 7
;"R";AT 13,28;"SUT"
5230 PRINT AT 15,3; INK 5; PAPER
 0;"C[redacted]D";AT 14,4; PAPER 7; INK
0; BRIGHT 1;"  ";AT 13,4; INK 3;
"GG"
5240 PRINT AT 5,8; INK 7; PAPER
0;"BBBBB";AT 6,8;"AAAAA";AT 4,8;
 INK 6;"C[redacted]D";AT 3,9; PAPER 7; I
NK 0; BRIGHT 1;"  ";AT 2,9; INK
3;"GG"
5242 RETURN
5245 REM PANT 5
5246 PRINT AT 8,5; INK 1; PAPER
7;"BBBBBBB": FOR N=1 TO 8: PRINT
 AT N,5; INK 1; PAPER 0;"A";AT N
,25;"A": NEXT N: PRINT PAPER 0;
INK 0;AT 6,25;" ";AT 4,25;" ";AT
 5,25;" ": FOR N=8 TO 15: PRINT
INK 5; PAPER 0;AT N,15;"A": NEXT
 N: PRINT AT 8,15; INK 5;"AAAAAA
AAAA": FOR N=4 TO 8: PRINT INK 5
;AT N,12;"B": NEXT N
5247 PRINT AT 13,2; PAPER 0; INK
 6;"BBBBB";AT 14,2;"AAAAA";AT 12
,3; INK 4; PAPER 0;"C[redacted]D";AT 11,
4; INK 3; PAPER 7; BRIGHT 1;"GG"
: PRINT AT 7,7; INK 7; PAPER 0;"
C[redacted]D";AT 6,8; PAPER 7; INK 0; BR
IGHT 1;"  ";AT 5,8; INK 3;"GG"
5248 PRINT AT 13,18; PAPER 0; IN
K 5;"BBBBBBBBBB";AT 14,18;"AAAAA
AAAAA";AT 12,19; INK 2; PAPER 0;
"C[redacted]D";AT 11,20; INK 3; PAPER 7;
 BRIGHT 1;"GG": PRINT AT 12,23;
INK 7; PAPER 0;"C[redacted]D"
5249 PRINT AT 11,8; INK 5; PAPER
 0;"NONO";AT 12,8;"QRQR";AT 13,8
;"QRQR";AT 14,8;"STST"
5250 PRINT AT 7,15; INK 1; PAPER
 7;"BBBBBBBBBB": PRINT AT 3,16;
INK 6; PAPER 0;"NONONO";AT 4,16;
"QRQRQR";AT 5,16;"QRQRQR";AT 6,1
6;"STSTST"
5999 RETURN
```

```
6000 REM presentacion
6400 RESTORE 6410
6410 DATA 231,153,165,90,90,165,
153,231
6415 FOR n=0 TO 7: READ q: POKE
USR "r"+n,q: NEXT n
6500 REM inicializar y marcadore
s y anagrama
6570 PRINT AT 0,0; PAPER 0; INK
4;"RRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRR"
6571 PRINT PAPER 7;AT 9,3;"ESPER
A, GENERANDO GRAFICOS"
6572 PRINT AT 17,0; PAPER 0; INK
 4;"RRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRR"
6575 FOR n=0 TO 17: PRINT PAPER
0; INK 4;AT n,0;"R";AT n,31;"R":
 NEXT n
6800 REM inicializar
6810 LET total=33: LET tar=0: LE
T pun=2: LET f$="J": DIM w(2): L
ET w(1)=-1: LET w(2)=1: LET j=0:
 LET k$="AB": LET q$="CDCE"
6820 LET v=1: LET pan=1: LET g=1
: DIM i(5): LET i(1)=8: LET i(2)
=11: LET i(3)=2: LET i(4)=6: LET
 i(5)=6: LET l=1: LET c=1: LET o
=1: LET p=1
6830 DIM s(2): LET s(1)=-1: LET
s(2)=2
6870 LET h=12: LET z=12: LET x=1
: LET y=1: LET a=1: LET b=1
6880 PRINT AT 1,2; INK 0; PAPER
0;"glotoh"
6885 LET e=2: LET s=30: FOR N=16
7 TO 160 STEP -1: FOR u=17 TO 64
6886 IF POINT (u,N) THEN PLOT e,
s: DRAW 3,0: PLOT e,s-2: DRAW 3,
0
6887 LET e=e+4
6888 NEXT u
6889 LET e=2: LET s=s-3: NEXT N
6905 PRINT AT 18,25; INK 3; PAPE
R 0;"PUNTOS";AT 19,27; INK 6;"33
";AT 20,24; INK 4;"PANTALLA";AT
21,27; INK 6;"1"
6910 PRINT AT 19,0; OVER 1; BRIG
HT 1; INK 3; PAPER 0;"
                 ";AT 20,0; INK 5;"
                          ";AT 18,0
; INK 4;"
"
6912 PRINT AT 21,7; INK 2; BRIGH
T 1;"© A.MARANON.": PAPER 0: INK
 0
7000 RESTORE 7100
7050 POKE 23675,88: POKE 23676,2
55: FOR n=0 TO 103: READ i: POKE
 USR "a"+n,i: NEXT n
7100 DATA 3,7,8,8,9,7,3,131,192,
224,224,160,224,208,104,104
7110 DATA 252,223,195,193,0,0,0,
0,104,232,210,196,226,117,50,32
7120 DATA 104,232,210,200,225,11
4,48,33,192,224,16,16,144,224,19
2,193
7130 DATA 3,7,7,5,7,11,22,22,63,
251,195,131,0,0,0,0,22,23,75,35,
71,174,76,4
7140 DATA 0,0,80,8,80,10,133,42
7160 DATA 241,138,76,62,25,105,1
01,131
7165 DATA 22,23,75,19,7,78,172,4
,0,0,0,10,16,74,161,64
7170 REM ***************************
7180 POKE 23675,8: POKE 23676,25
4
7190 RESTORE 7200: FOR n=0 TO 16
7: READ e: POKE USR "a"+n,e: NEX
T n
7200 DATA 85,110,85,203,54,42,54
,28
7210 DATA 85,170,85,170,85,170,8
5,170
7220 DATA 255,192,127,127,63,63,
31,15,255,3,254,254,252,252,248,
240,6,9,9,6,0,0,0,0,24,36,102,15
3,153,102,36,24
7230 DATA 255,255,118,52,36,98,1
78,224
7240 DATA 7,61,254,143,240,127,9
5,47,224,124,171,227,31,254,254,
252,23,11,5,3,1,1,1,1,248,240,22
4,192,128,128,128,128,1,1,1,1,6,
11,22,63,128,128,128,0,224,240,2
48,252
7250 DATA 0,15,16,38,64,255,192,
127,0,240,8,4,2,255,129,255,24,2
55,0,0,0,255,0,255
7255 DATA 80,80,80,64,80,80,80,8
0,2,2,2,2,2,2,2,2,64,80,80,80,64
,79,80,127,2,10,10,10,2,242,10,2
54,0,0,0,0,0,255,0,255
7300 PAUSE 200: GO SUB 7700
7320 PRINT PAPER 7;AT 9,8;" PULS
A UNA TECLA ": PAUSE 0: GO TO 77
00
7600 RETURN
7700 FOR N=1 TO 16: PRINT PAPER
0; INK 0;AT N,1;"
                ": NEXT N
7710 RETURN
8000 PRINT AT 8,8; FLASH 1; INK
7; PAPER 0;"EXTERMINIO TOTAL"
8010 LET KL=INT (RND*20): FOR n=
1 TO 121: READ F: BEEP .2,F+KL:
BEEP .06,F-10: NEXT n: BEEP 1,11
8020 PAUSE 0: GO SUB 7700: GO TO
 1
8100 REM fin tarta pantalla
8110 FOR n=1 TO 30: BEEP .2,INT
(RND*40): PRINT AT 10,10; PAPER
0; INK INT (RND*7)+1;"HAS FRACAS
ADO": NEXT n: FOR N=1 TO 15: PRI
NT PAPER 0; INK 0;AT N,1;"
                         ": NEXT
N: GO TO 6800
```

AMSTRAD CPC-464

AMSTRAD



ORDENADORE

SERIE CPC

UNIDAD CENTRAL. MEMORIAS
• Microprocesador Z80A - 64K RAM ampliables - 32K ROM ampliables **• TECLADO •** Teclado profesional con 74 teclas en 3 bloques - Hasta 32 teclas programables - Teclado redefinible **• PANTALLA •** Monitor RGB verde (12") o color (14")

	Normal	Alta Res.	Multicolor
Col × líneas	40 × 25	80 × 25	20 × 25
Colores	4 de 27	2 de 27	16 de 27
Puntos	320 × 200	640 × 200	160 × 2

— Se pueden definir hasta 8 ventanas de texto y 1 de gráficos **• SONIDO**
• 3 canales de 8 octavas moduladas independientemente - Altavoz interno regulable - Salida estéreo **• BASIC**
• Locomotive BASIC ampliado en ROM - Incluye los comandos AFTER y EVERY para control de interrupciones

AMSTRAD CPC 464

CASSETTE • Cassette incorporada con velocidad de grabación (1 ó 2 Kbaudios) controlada desde Basic **• CONECTORES**
• Bus PCB multiuso, Unidad de Disco exterior, paralelo Centronics, salida estéreo, joystick, lápiz óptico, etc.
• SUMINISTRO • Ordenador con monitor verde o color - 8 cassettes con programas - Libro "Guía de Referencia BASIC para el programador" - Manual en castellano - Garantía Oficial AMSTRAD ESPAÑA.

TODO POR	**59.900 Pts.**	(monitor verde)
	90.900 Pts.	(monitor color)

AMSTRAD CPC 6128

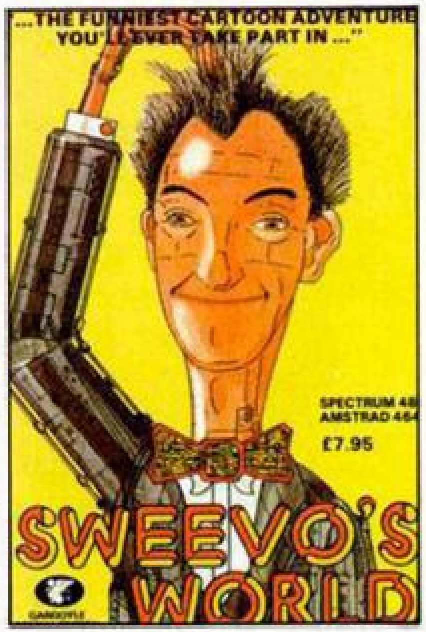
UNIDAD DE DISCO • Unidad incorporada para disco de 3" con 180K por cara **• SISTEMAS OPERATIVOS**
• AMSDOS, CP/M 2.2, CP/M Plus (3.0)
• CONECTORES • Bus PCB multiuso, paralelo Centronics, cassette exterior, 2.ª Unidad de Disco, salida estéreo, joysticks, lápiz óptico, etc.
• SUMINISTRO • Ordenador con monitor verde o color - Disco con CP/M 2.2 y lenguaje DR. LOGO - Disco con CP/M Plus y utilidades - Disco con 6 programas de obsequio - Manual en castellano - Garantía Oficial AMSTRAD ESPAÑA.

TODO POR	**84.900 Pts.**	(monitor verde)
	119.900 Pts.	(monitor color)

SWEEVO'S WORLD • Videoaventura • Gargoyle Games

LAS ANDANZAS DEL ROBOT «SWEEVO»

Primero fue Tir Na Nog, después Dun Darach y ahora, tras su último éxito, Marsport, Gargoyle Games vuelve a la carga con una nueva y simpática videoaventura: Sweevo's World.



Gargoyle Games ha roto completamente con la tradición. Atrás queda ya la bella ciudad de Tir Na Nog, por la que el recio y melenudo Cuchulain dejará de pasear por algún tiempo.

Aquellos eran buenos tiempos, pero como bien dice el refrán: «Renovarse o morir». Y esto es lo que Gargoyle acaba de hacer. Tras realizar una última tentativa de éxito con Marsport, programa éste como ya sabréis totalmente inspirado en Dun Darach, ha dado un giro a su estilo de programación y ha realizado este curioso Sweevo's World.

Esta nueva videoaventura está muy en la línea de las obras de Ultimate (Knight Lore o Alien 8), y además de utilizar la ya clásica técnica Filmation, el juego tiene una filosofía muy parecida a la de éstos, así como la perspectiva y presentación general de las pantallas, que también recuerdan, quizás excesivamente, a la de dichos programas.

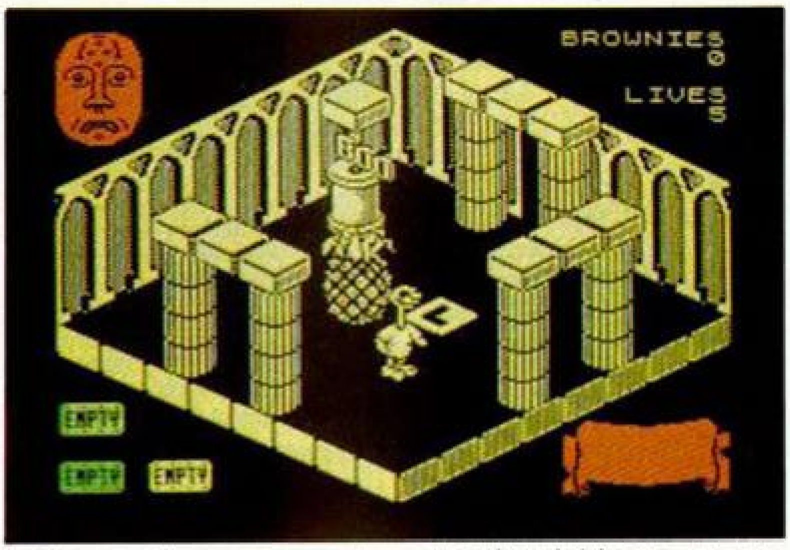
Sin embargo, todo es mejorable, y Sweevo's World, es un programa cargado de originalidad e imaginación, que, a pesar de estar basado en estos puntos genéricos, aporta bastantes características de creación propia.

El argumento consiste en un despistado y torpe robot, de aspecto bastante parecido al de E. T., el cual debe resolver una complicada tarea de recolección de objetos.

Sweevo debe recorrer cuatro mundos diferentes, los cuales se encuentran uno encima del otro y que se comunican entre sí gracias a elevadores aéreos y agujeros gravitatorios. Así, nuestro feo protagonista podrá pasear a su antojo por las diferentes estancias de Pino solitario, Realmentelibre, Dedos y Pastel de manzana, pues éstos son los curiosos nombres de los mundos.

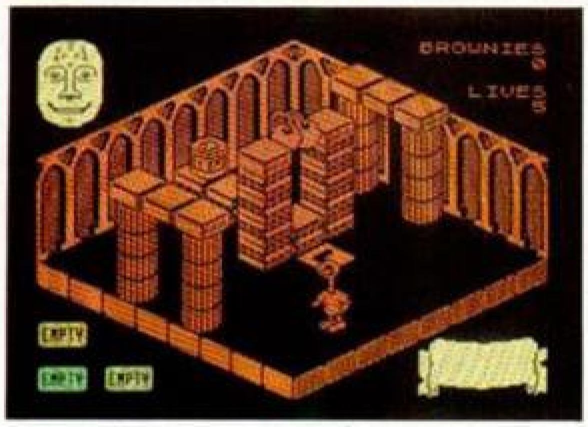
Allí deberá encontrar cuatro tipos diferentes de objetos: botas, ositos de peluche, latas de comida y del mismo modo, deberá intentar rescatar el mayor número posible de Brownies (pequeños seres de gran cabeza que permanecen la mayor parte de su vida sentados).

Pero, por supuesto, esto no sería un juego si la cosa sólo fuera llegar y coger los objetos sin la menor complicación. Así, las habitaciones se encuentran completamente infectadas de ex-

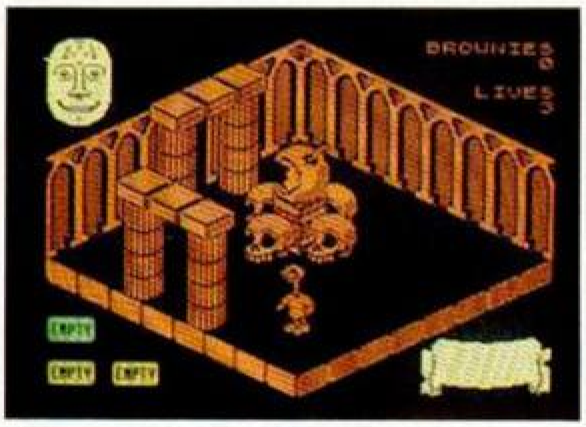


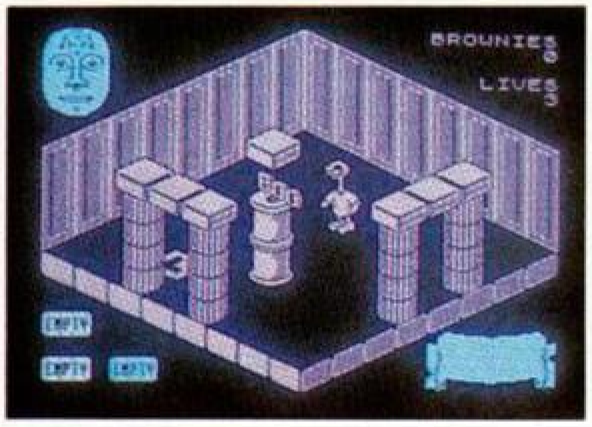
traños y peligrosos enemigos y obstáculos que nos dificultarán, como es de suponer, nuestra entretenida labor. Sin embargo, en este detalle reside el mayor atractivo del juego, ya que estos impedimentos son los que realmente hacen que todo resulte divertido, debido a la gran variedad de los mismos (dedos que sur-

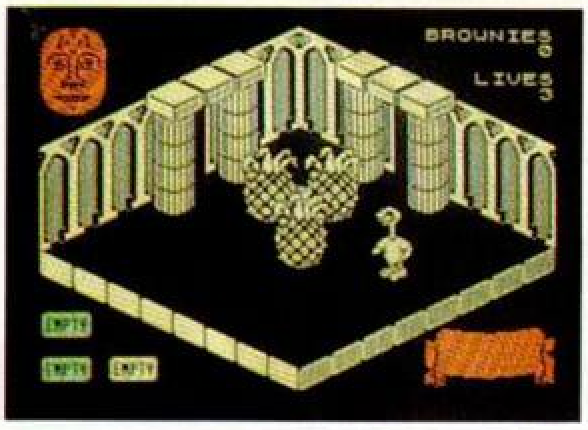


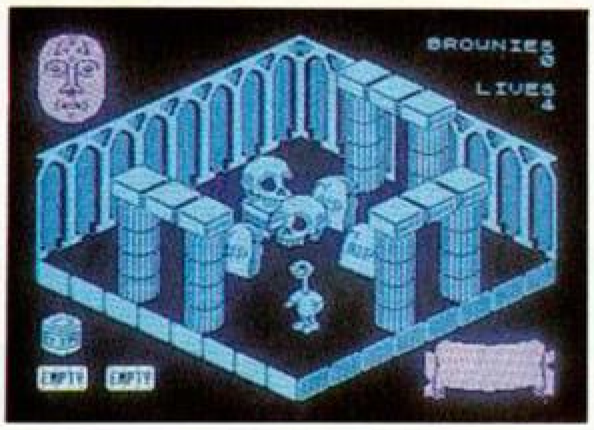












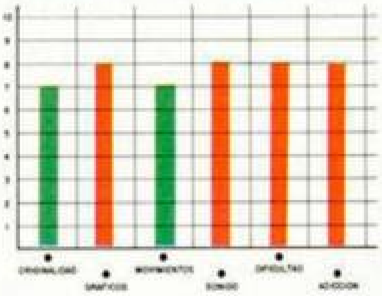
gen repentinamente del suelo, enormes pesas que caen del techo, frutas venenosas, enanos traviesos, losas móviles, enormes calaveras...).

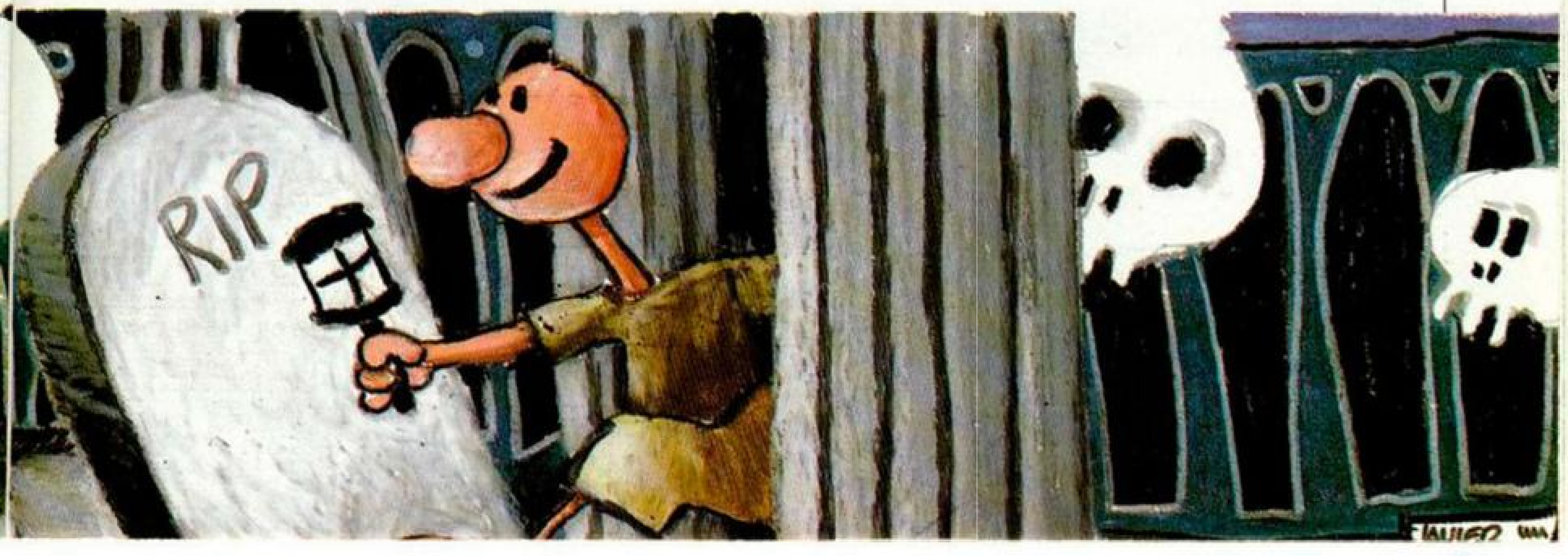
De este modo, el juego se hace bastante entretenido, ya que además de habilidad para ir sorteando los peligros que nos asalten por sorpresa, necesitaremos también pensar una táctica para librarnos de ciertos obstáculos que nos impedirán llegar hasta el lugar en el que se encuentran los objetos, por lo que de alguna manera se puede decir que muchas de las pantallas tienen la típica estructura de un arcade.

Y volviendo al tema de los gráficos de este Sweevo's World, tenemos que recalcar la gran imaginación con la que han sido realizados cada uno de los múltiples detalles que configuran las diferentes pantallas, las cuales resultan por ello, bastante atractivas y de una excelente calidad.

Otra característica que llama la atención, pues muy pocas veces hemos podido ver algo parecido, es la manera tan particular que posee el programa para indicarnos cómo anda nuestro personaje de energía y fuerzas. Esto se realiza mediante una enorme cara que aparece en la parte superior de la pantalla, la cual va cambiando tanto de color como de expresión (de más alegre a más triste), a medida que vayan disminuyendo sus constantes vitales.

En resumen, podemos decir que tanto por los aspectos visuales de Sweevo's World, como por lo divertido y entretenido que resulta, se trata de un excelente programa con el que Gargoyle Games volverá a situarse de nuevo en la lista de las más prestigiosas casas de software, como viene siendo habitual con sus más recientes creaciones.





Deportivo. Imagine-Konami

PING-PONG



Un nuevo juego viene a sumarse a la ya larga lista de programas de simulación deportiva, dispuesto a hacernos participar de toda la emoción de la competición. Desde el baloncesto al tenis, desde el atletismo al kárate, pasando por deportes tan dispares como la equitación, el piragüismo o el patinaje artístico, pueden ser vividos intensamente y sin apenas esfuerzo, con el simple hecho de sentarnos frente a un pequeño ordenador.

Y, por supuesto, en esta larga lista no podría faltar un deporte que con su gran emoción y trepidante acción, puede hacer las delicias de todos: el ping-pong.

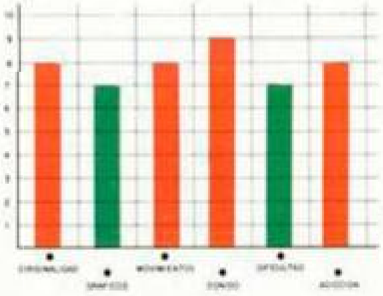
Así, nada más cargar el programa, nos encontramos ante *una larga mesa*, raqueta en mano y pelota en ristre dispuestos a comenzar inmediatamente a derrotar a nuestro contrincante, quien, por el momento, permanece impasible ante nosotros. No obstante, tras botar *ligeramente la bola*, asestamos un preciso saque que le obliga a reaccionar rápidamente y consigue devolver el golpe. Un severo revés sale de nuestra pala que deja desconcertado al rival quien, sin embargo, consigue enviar «in extremis» la pelota a nuestro campo. Al llegar a media altura, golpeamos la pelota con fuerza, efectuando un sobervio mate con el cual culminamos la jugada y despertamos la admiración del público, quién estalla en un sonoro aplauso.

Efectivamente, así de emocionante y divertido resulta este Ping-pong, al cual se ha conseguido imprimir un gran realismo gracias a la gran cantidad de golpes distintos que podemos llevar a cabo y que hacen que podamos enviar la bola prácticamente en cualquier dirección o con diferentes clases de efectos.

Sin embargo, al principio resulta bastante difícil el llegar a dominar perfectamente todos los movimientos que se pueden realizar con la raqueta, por lo que tardaremos algún tiempo en hacernos verdaderos expertos en tan bello deporte.

En cuanto a los aspectos *gráficos y de movimiento*, están bastante bien cuidados, pero sobre todo llama la atención la sensacional melodía que acompaña al juego, la cual se puede considerar de lo mejorcito oído en un micro-ordenador.

En fin, un buen juego.



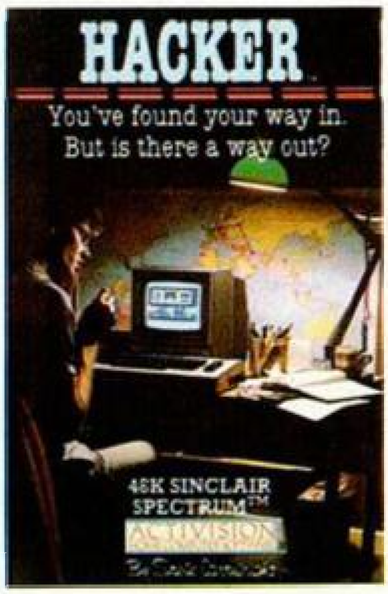




Hacker. Estrategia. Activision

UNA INCOGNITA ATRACTIVA

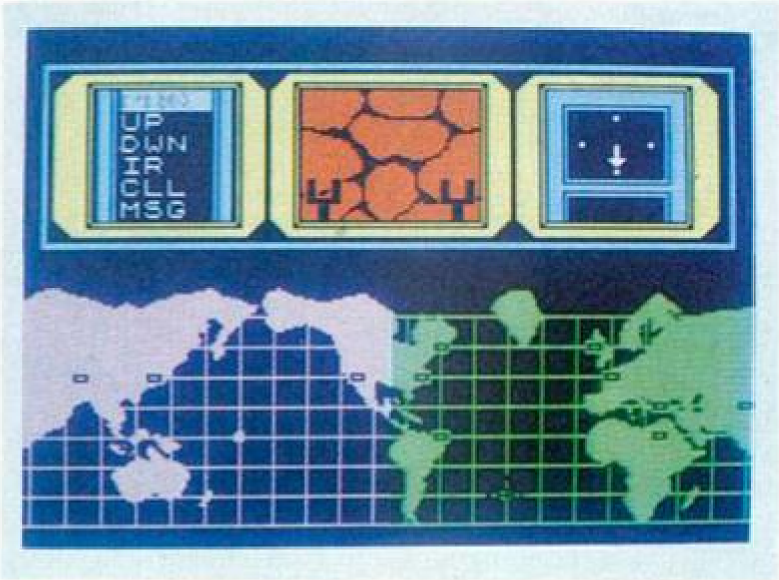
De todos es conocido, aunque sólo sea de una manera intuitiva, el significado de la palabra Hacker. Según la traducción literal, el Hacker es una especie de mercenario que lucha denodadamente por resolver todas las dificultades que se le plantean para conseguir una misión determinada. Este término aplicado a la informática, nos sugiere la idea de un auténtico «jugón», cuyo principal objetivo es el de resolver los secretos de un programa y llegar a adentrarse en sus rincones más escondidos.

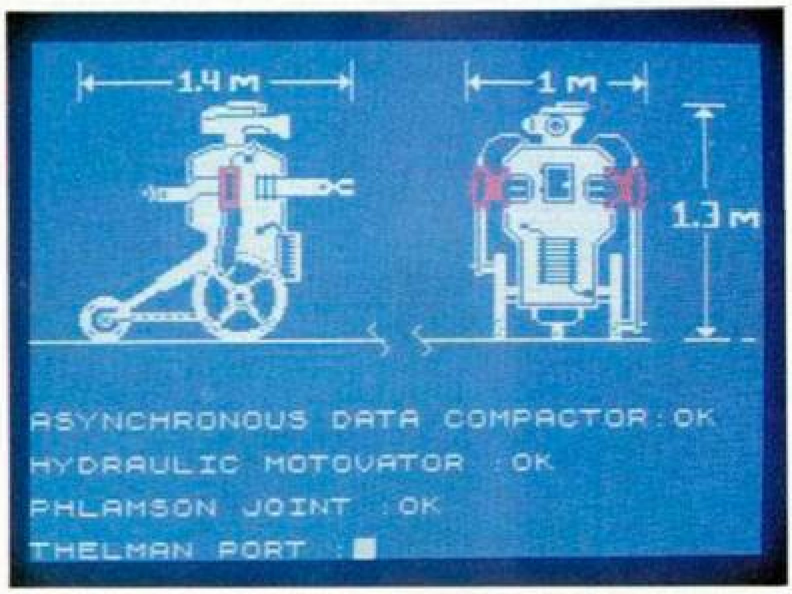


Hacker es un claro ejemplo de qué es lo que un mercenario del software debe hacer; y la verdad es que nos pone las cosas bastante difíciles.

De entrada, no sabemos ni de qué trata el juego, ni por dónde hay que empezar ni qué es lo que tenemos que conseguir. En una palabra: nada. Ni siquiera las propias instrucciones nos ofrecen ningún tipo de pista que nos pueda servir de orientación, sino que, por el contrario, nos ponen sobre aviso de que seremos nosotros quienes tendremos que ir descubriendo como podamos, el desarrollo de las distintas fases del juego.

Por esta misma razón, no



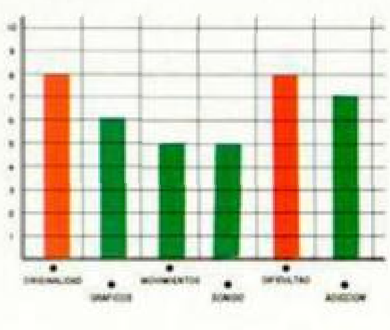


sería demasiado adecuado el que nosotros os desveláramos lo que hemos llegado a descubrir con mucho esfuerzo y buena voluntad, puesto que sería ir un poco en contra del propio interés del juego.

Sin embargo, sí os podemos decir que Hacker es, por supuesto, un juego de estrategia, en el cual debemos mantener un diálogo casi constante con el ordenador, durante el transcurso del cual, se nos harán un buen número de preguntas y se nos pedirá que introduzcamos diferentes tipos de claves. Estas, evidentemente, no aparecen escritas en ningún lugar y tendremos que ser nosotros los que con mucha paciencia y haciendo uso de nuestra brillante inteligencia, vayamos encontrando los caminos necesarios para ir avanzando en el juego.

En cuanto a sus aspectos gráficos, no tienen una presentación excesivamente cuidada, pero esto responde al hecho de que Hacker no es un juego dedicado a satisfacer las exigencias visuales, sino más bien las mentales, aunque tampoco se puede decir que el aspecto estético del programa sea deficiente.

Hacker, en definitiva, es un juego en el que si se logra superar las primeras dificultades, puede convertirse en algo bastante adictivo e interesante.

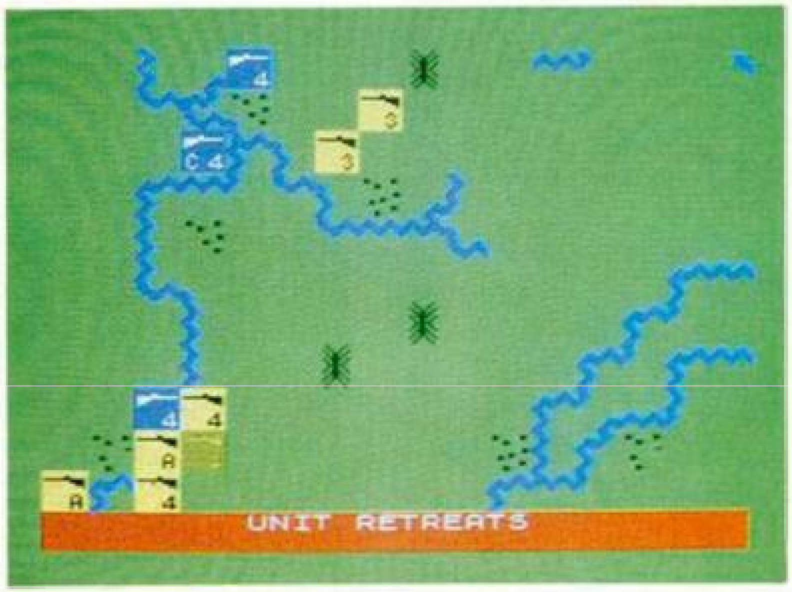


Austerlitz. War-game. Lothlorien.

LA VICTORIA NAPOLEONICA

Más war-games. En esta ocasión presentado por Lothlorien, la casa más prestigiosa en este tipo de programas, cuya merecida fama ha sido lograda con juegos como Special Operations, Waterloo, The Bulge, y ahora con este excelente Austerlitz.

Austerlitz es un war-game diseñado para un solo jugador, en el cual se nos traslada al crudo invierno de 1805 y se nos coloca al mando de los ejércitos napoleónicos dispuestos a enfrentarnos, en una cruenta y durísima batalla, a las tropas austro-rusas, comandadas por el zar Alejandro de Rusia y el emperador Francisco de Austria.

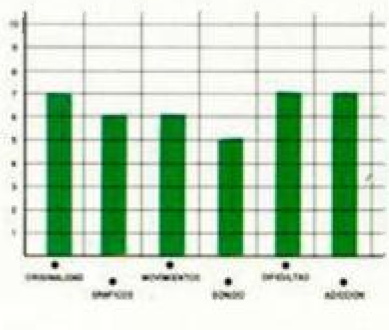


La historia nos cuenta cómo Napoleón, con sus asombrosas dotes de estratega consiguió derrotar a sus enemigos con suma facilidad, gracias a la excelente preparación de su ejército, notablemente inferior en número al de las tropas aliadas.

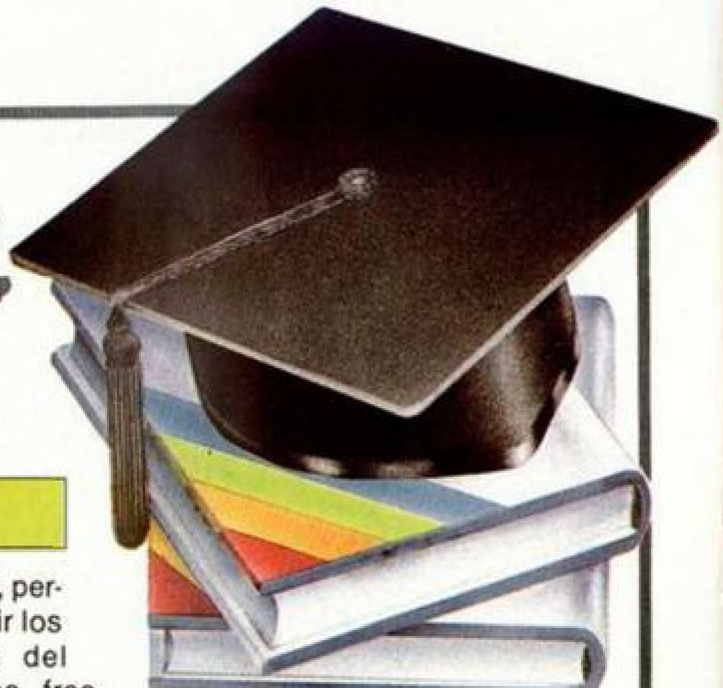
En el desarrollo del juego, la contienda está completamente equilibrada, puesto que nuestro oponente dispone de idéntica información bélica que nosotros, y el resultado de la batalla dependerá exclusivamente de factores como la fuerza, el terreno, la moral y, por supuesto, de la suerte.

Uno de los detalles más destacados de este Austerlitz que, dicho sea de paso, posee las mismas características gráficas que el resto de programas de la misma casa, es el de que consigue un gran realismo debido a la existencia en nuestro ejército de cuatro comandantes con voluntad propia. Es decir, que como ocurriría en un auténtico combate, debemos dar las instrucciones pertinentes a nuestros subordinados, pero luego ellos podrán hacer su propia interpretación de las mismas.

Un buen juego de estrategia, como siempre, correctamente realizado y muy interesante y bastante entretenido.



Profesor particular



Arturo LOBO y J. J. LEON

SISTEMAS ELEMENTALES

El programa de esta semana va dirigido a los estudiantes de octavo curso de EGB, y pretende ser una ayuda en vuestro primer contacto con la resolución de sistemas lineales de 2 incógnitas.

Como ya sabréis, existen tres métodos fundamentales de resolución exacta: reducción, sustitución e igualación, además del método gráfico (asimilando ambas ecuaciones a rectas y representando su punto de corte). Con este programa podéis introducir el sistema y él os irá guiando, por el sistema que escojáis, hasta llegar a la solución.

Este programa tiene la particularidad de que maneja fracciones además de números enteros, con lo que resulta mucho más didáctico. Asimismo, permite introducir los coeficientes del sistema como fracciones: para ello debéis introducir «f», ENTER, numerador (con el signo), ENTER, denominador y ENTER, pero conviene señalar que, en todo caso, el sistema debe ser compatible y determinado puesto que, por no alargar el programa, no realiza esa comprobación por sí mismo. Para terminar una advertencia: las líneas DATA del final son una rutina máquina y no podéis cometer ninguna equivocación al teclearlas. Por si acaso, os recomendamos grabar el programa antes de hacerlo correr.

```
   2 CLEAR 64999: GO SUB 8000: RESTORE : DIM a(2)
   3 PAPER 6: BORDER 6: CLS
 110 CLS : PRINT AT 3,12;"MENU",AT 6,4;
 120 RESTORE 150: FOR i=1 TO 4
 130 READ a$: PRINT a$''TAB 4;: BEEP .1,1: NEXT i
 150 DATA "1)INTRODUCIR ECUACIONES","2)METODO DE SUSTITUCION","3)METODO DE REDUCCION","4)METODO DE IGUALACION"
 160 LET a$=INKEY$
 170 IF CODE a$>=49 AND CODE a$<=52 THEN LET menu=VAL a$: GO TO 1000*menu
 180 GO TO 160
 210 FOR i=0 TO 1: LET f12=0: GO SUB 215: PRINT ''': NEXT i: RETURN
 215 LET j=1
 220 LET a(1)=z(3*i+j,1)
 230 LET a(2)=z(3*i+j,2)
 240 LET a=a(1)/a(2): IF NOT a THEN GO TO 290
 250 IF j=2 THEN PRINT "+";: PRINT ((CHR$ 8) AND a<0);
 260 GO SUB 7000: PRINT ((CHR$ 8) AND ABS a=1);
 280 PRINT CHR$ (119+j);: GO SUB 350
 290 IF j=1 THEN LET j=2: GO TO 220
 300 PRINT "=";: LET a(1)=z(3*(i+1),1): LET a(2)=z(3*(i+1),2)
 310 GO SUB 7000: RETURN
 350 IF f12 THEN IF jj=j THEN PRINT CHR$ 8;"(";: GO SUB 600: PRINT ")";
 360 RETURN
 410 LET a1=(ABS a(1) AND ABS a(1)>=ABS a(2))+(ABS a(2) AND ABS a(2)>ABS a(1))
 420 LET a2=ABS a(1)+ABS a(2)-a1
 425 IF NOT a2 THEN RETURN
 430 LET rr=a1-a2*INT (a1/a2)
 440 IF rr=0 THEN LET a(1)=a(1)/a2: LET a(2)=a(2)/a2: RETURN
 450 LET a1=a2: LET a2=rr: GO TO 430
 500 PRINT #0;AT 1,8;"Pulsa una tecla": PAUSE 0: PRINT #0;AT 1,8;TAB 24;: RETURN
 600 LET a(1)=z(3*ii+3,1)*z(3*ii+jj,2)
 610 LET a(2)=z(3*ii+3,2)*z(3*ii+jj,1): LET f11=0
 620 IF a(1) THEN GO SUB 400: GO SUB 7000: PRINT "+";: LET f11=1
 630 IF NOT z(3*ii+3-jj,1) THEN PRINT CHR$ 8;" ";: GO TO direccion
 640 LET a(1)=-z(3*ii+3-jj,1)*z(3*ii+jj,2)
 650 LET a(2)=z(3*ii+3-jj,2)*z(3*ii+jj,1)
 660 IF a(1)<0 AND f11 THEN PRINT CHR$ 8;
 670 GO SUB 400: GO SUB 7000: PRINT ((CHR$ 8) AND ABS (a(1)*a(2))=1);CHR$ (122-jj);
 680 RETURN
 800 FOR i=1 TO 6: FOR j=1 TO 2: LET a(j)=z(i,j): NEXT j: GO SUB 400: LET a(2)=a(2) OR NOT a(1): FOR j=1 TO 2: LET z(i,j)=a(j): NEXT j: NEXT i: RETURN
1010 CLS : PRINT AT 0,0;"ax+by=c"''"dx+ey=f"
1020 PRINT AT 6,0;"Introduce los valores de a,b,c, d,e y f"''"Si alguno es fraccionario intro-duce antes la letra ""f"""
1030 DIM z(6,2): FOR i=1 TO 6
1040 INPUT CHR$ (96+i);"? ";LINE a$
1050 IF a$<>"f" THEN LET z(i,1)=VAL a$: LET z(i,2)=1: GO TO 1070
1060 INPUT CHR$ (96+i);"? ";z(i,1);"/";z(i,2)
1070 NEXT i: CLS : PRINT '': GO SUB 200
1080 GO SUB 500: CLS
1090 RESTORE 1150: DIM x(3,2)
1100 FOR m=1 TO 3: READ p,q,r,s
1110 LET a(1)=z(p,1)*z(q,1)*z(r,2)*z(s,2)-z(r,1)*z(s,1)*z(p,2)*z(q,2)
1120 LET a(2)=z(p,2)*z(q,2)*z(r,2)*z(s,2)
1130 GO SUB 400: LET x(m,1)=a(1): LET x(m,2)=a(2): NEXT m
1140 DIM y(2,2): FOR m=1 TO 2: LET a(1)=x(m+1,1)*x(1,2): LET a(2)=x(m+1,2)*x(1,1): GO SUB 400: LET y(m,1)=a(1): LET y(m,2)=a(2): NEXT m
1150 DATA 1,5,4,2,3,5,2,6,1,6,3,4
1160 GO TO 100
2000 LET direccion=2150
2010 CLS : PRINT : GO SUB 200
2020 FOR i=0 TO 1: FOR j=1 TO 2
2030 IF z(3*i+j,1)=0 THEN LET j=3-j: GO TO 2090
2040 NEXT j: NEXT i
2050 FOR i=0 TO 1: FOR j=1 TO 2
2060 IF z(3*i+j,1)/z(3*i+j,2)=1 THEN GO TO 2090
2070 NEXT j: NEXT i
2080 LET i=INT (RND*2): LET j=INT (RND*2)+1
2090 LET ii=i: LET jj=j
2100 PRINT AT 8,0;"De la ecuacion ";ii+1;" tenemos:"
2110 PRINT AT 10,10;CHR$ (119+jj);"=";
2120 GO SUB 600: GO SUB 500
2125 IF menu<>2 THEN RETURN
2130 PRINT AT 12,0;"Sustituyendo en la ecuacion ";2-ii;":": LET f12=1: LET i=NOT ii
2140 PRINT AT 14,10;: GO SUB 215: GO SUB 500
2150 PRINT AT 17,0;"Operando: ";
2160 GO TO 6500
3000 CLS : PRINT : GO SUB 200: LET direccion=a
3010 DIM w(2,2): LET jj=1+(z(1,1)=0 OR z(4,1)=0)+(ABS (z(2,1)/z(2,2)*z(5,2)/z(5,1))=1)
3015 DIM v(6,2): FOR i=1 TO 6: FOR j=1 TO 2: LET v(i,j)=z(i,j): NEXT j: NEXT i
3020 FOR i=0 TO 1: LET a(i+1)=z(jj+3*i,1)/z(jj+3*i,2): NEXT i
3030 GO SUB 400: FOR i=1 TO 2: FOR j=1 TO 2
3040 LET w(i,j)=z(jj+3*(i-1),j): IF i=1 AND j=1 THEN LET w(i,j)=-w(i,j)
3050 NEXT j: NEXT i
3060 IF a2=INT a2 THEN FOR i=1 TO 2: LET w(i,2)=w(i,2)*a2: NEXT i
3070 PRINT AT 6,0;"Multiplicando la ecuacion 1 por ";: LET in=2
3080 GO SUB 3500: PRINT " y la 2 por ";: LET in=1: GO SUB 3500: PRINT '" queda:"''
3085 FOR r=1 TO 2: FOR s=1 TO 2: FOR t=1 TO 3: LET z(t+(3 AND s=1),r)=z(t+(3 AND s=1),r)*w(s,r): NEXT t: NEXT s: NEXT r: GO SUB 800: GO SUB 200
3090 GO SUB 500: PRINT AT 13,18;"Y sumando:"''
3100 LET a(1)=z(3,1)*z(6,2)+z(3,2)*z(6,1): LET a(2)=z(3,2)*z(6,2): GO SUB 400: DIM u(2)
3110 FOR s=1 TO 2: LET u(s)=a(s): LET a(s)=a(s)*y(3-jj,3-s): NEXT s
3120 PRINT AT 15,18;: GO SUB 400: GO SUB 7000: PRINT CHR$ (122-jj);"=";: LET a(1)=u(1): LET a(2)=u(2): GO SUB 7000
3130 GO SUB 500: LET ii=0: FOR i=1 TO 6: FOR j=1 TO 2: LET z(i,j)=v(i,j): NEXT j: NEXT i: GO TO 4050
3500 LET a(1)=w(in,1): LET a(2)=w(in,2): GO SUB 400: GO SUB 7000: RETURN
4000 LET direccion=4050
4010 GO SUB 2010
4020 PRINT AT 12,0;"Y de la ecuacion ";2-ii;":": LET ii=NOT ii
4030 PRINT AT 14,10;CHR$ (119+jj);"=";: GO SUB 600: GO SUB 500
4040 LET ii=NOT ii: PRINT AT 16,0;"Igualando: ";: GO SUB 600: PRINT "=";: LET ii=NOT ii: GO SUB 600: GO SUB 500
4050 PRINT AT 18,0;"De donde ";
4060 GO TO 6500
6500 PRINT CHR$ (122-jj);"=";: LET a(1)=y(3-jj,1): LET a(2)=y(3-jj,2): GO SUB 7000
6510 PRINT '"Sustituyendo en ec. ";ii+1;" ";CHR$ (119+jj);"=";
6520 LET a(1)=y(jj,1): LET a(2)=y(jj,2): GO SUB 7000: GO SUB 500: GO TO 100
7000 IF ABS a(2)=1 THEN PRINT a(1)*SGN a(2);: GO TO 7130
7005 PRINT (("-") AND a(1)*a(2)<0);: LET a(1)=ABS a(1): LET a(2)=ABS a(2)
7010 LET a1=LEN STR$ a(1): LET a2=LEN STR$ a(2)
7020 LET len=a1+(a2-a1)*(a2>a1)
7030 DIM c$(len): DIM b$(len)
7040 LET c$(len-a1+1 TO len)=STR$ a(1): LET b$(len-a2+1 TO len)=STR$ a(2)
7050 LET p2=-21+8*PEEK 23689
7060 FOR l=0 TO len-1
7070 LET p1=(33-PEEK 23688)*8+1+6*l
7080 PLOT p1,p2: DRAW 6,0
7090 IF a2>=len-l THEN LET a3=VAL b$(l+1): RANDOMIZE FN p(p1+1-3*(len-a2),p2-7,a3)
7100 IF a1>=len-l THEN LET a4=VAL c$(l+1): RANDOMIZE FN p(p1+1-3*(len-a1),p2+1,a4)
7110 NEXT l
7120 FOR l=1 TO INT ((-1+6*len)/8)+1: PRINT OVER 1;" ";: NEXT l
7130 RETURN
8000 RESTORE 8050
8010 FOR i=65000 TO 65077: READ a: POKE i,a: NEXT i
8020 FOR i=65400 TO 65431: READ a: POKE i,a: NEXT i
8030 FOR i=65400 TO 65527 STEP 4: FOR j=0 TO 2: READ a: POKE i+j,a: NEXT j: POKE i+3,0: NEXT i
8050 DATA 205,148,30,245,205,153,30,197,205,153,30,197,225,209,123,198,5,95,241,229,30,255,203
8060 DATA 39,203,39,198,200,111,78,44,86,44,70,225,96,121,205,120,255,205,120,255,122,205,120
8070 DATA 255,205,120,255,124,205,120,255,125,198,5,6,0,79,205,43,45,1,0,0,205,43,45,1,1,0,197,205,43,45,193,201
8080 DATA 6,4,197,203,127,40,11,67,77,245,213,229,205,223,34,225,209,241,44,203,7,193,16,234,79,125,214,4,111,121,29,201
8090 DATA 105,153,96,36,34,32,105,36,240,105,41,96,36,111,32,240,225,224,104,233,96,241,36,64,105,105,96,105,113,96,0,0,32,0,112,0
8100 DEF FN p(x,y,n)=x+y*n+USR 65000
8110 RETURN
```

Por último, hemos utilizado un nuevo pseudo-nemónico: «DEFM» que significa: «DEFine Mesage» es decir, definir un mensaje. Lo que hace el ensamblador cuando se encuentra con este pseudo-nemónico es colocar en los bytes siguientes los códigos de los caracteres que componen el mensaje encerrado entre comillas.

Ahora ya, vamos a ver cómo funciona la rutina. En la línea 1010 cargamos en «HL» la dirección de inicio del mensaje. Luego, cargamos en «B» la longitud del mismo (en este caso, 20 bytes). A continuación, entramos en un bucle donde iremos cargando, uno a uno los bytes que componen el mensaje, en el registro «A», preservando «HL» y «BC» en la pila, llamando a la rutina «IMP-A», recuperando los registros, incrementando el puntero y cerrando el bucle hasta que se hayan impreso los 20 caracteres que componen el mensaje.

Para utilizarlo desde Basic, podemos hacer:

```
PRINT AT li,co;: RANDOMIZE USR 60500
```

Donde «li» y «co» son las coordenadas del punto a partir del cual deseamos imprimir el mensaje. Esta rutina se puede emplear para imprimir cualquier mensaje siempre que su longitud no exceda de 255 caracteres.

También es importante señalar que la rutina «IMP-A» lee los caracteres del font direccionado por la variable del Sistema «CHARS» de forma que, si hemos cargado otro juego de caracteres y lo tenemos direccionado mediante

```
60104 700          LD      D,A
      710
60105 720          LD      (DF_CC),DE
60109 730          POP     DE
60110 740          LD      HL,#1821
60113 750          SBC     HL,DE
60115 760          LD      (S_POSN),HL
60118 770          RET
23606 780 CHARS    EQU     23606
23684 790 DF_CC    EQU     23684
23681 800 BAND     EQU     23681
23688 810 S_POSN   EQU     23688
 3582 820 SCROLL   EQU     #0DFE
      990 ;
605001000 TEST     ORG     60500
605001010          LD      HL,MENS
605031020          LD      B,20
605051030 LOOP     LD      A,(HL)
605061040          PUSH    BC
605071050          PUSH    HL
605081060          CALL    IMP_A
605111070          POP     HL
605121080          INC     HL
605131090          POP     BC
605141100          DJNZ    LOOP
605161110          RET
605171120 MENS     DEFM    "curso C/M "
605271130          DEFM    "MICROHOBBY"


Pass 2 errors: 00


Table used:    192  from    246
```

Fig. 9-18. Listado completo de «IMP-A» y «TEST».

«CHARS», la rutina «IMP-A» usará, precisamente, esos caracteres. En la Figura 9-18 tiene el listado completo de las rutinas vistas hasta ahora.

Para indicarle a la rutina si ha de imprimir en normal, cursiva, negrita o ambas, lo haremos «POKEando» en la dirección 23681. Si escribimos un «0», la rutina imprimirá normalmente; con un «1» imprimirá letra cursiva; con un «2» la letra será negrita y, finalmente, con un «3» será cursiva y negrita a la vez.

Ya hemos llegado al punto donde toca ensamblar el programa. Esta vez, y para variar, lo ensamblaremos en hexadecimal en lugar de hacerlo en decimal como hasta ahora. Ensamblar en hexadecimal es exactamente igual de fácil (y trabajoso) que hacerlo en decimal. Además, tendremos la ventaja de que los «DATAs» del programa Basic que utilicemos ocuparán considerablemente menos memoria. Cada byte en decimal ocupa 5 octetos mientras que en Hexa ocupa 2. La diferencia sólo compensa para rutinas largas, ya que el programa cargador se complica bastante al tener que convertir los datos de hexadecimal a decimal. Luego veremos esto con más detalle, de momento, vamos a ensamblar.

Las dos instrucciones «CALL» que utilizamos no se han visto hasta ahora, así que

En las líneas 100 y 110, cogemos los tres bits inferiores del número de línea. En 120, 130 y 140 los rotamos a la derecha tres veces que equivale a rotarlos cinco veces a la izquierda —tenga en cuenta que los bits que salen por la derecha entran por la izquierda. En 150, los mezclamos con el número de columna y ése será el octeto bajo de nuestra dirección que, en la línea 160, almacenamos en «E».

Para componer el octeto alto, empezamos por cargar en «A» el número de línea y aislar (con AND 18) los dos bits que nos quedaban (líneas 170 y 180). En la línea 190, hacemos un «OR» con 40 para añadir la constante «010» al principio del octeto. Con esto, tenemos completo el byte alto de nuestra dirección; sólo queda almacenarlo en «D», lo que se hace en la línea 200. A la salida de la rutina, tendremos en «DE» la dirección de pantalla del octeto situado en el primer scan del carácter cuyas coordenadas contenía «DE» cuando entramos.

Con todos estos datos, ya podemos manejar la pantalla con bastante facilidad. En sucesivos capítulos, iremos viendo distintos efectos que se pueden conseguir sobre la pantalla. De momento, podemos hacer una rutina que nos permita imprimir un carácter en una determinada posición de la pantalla.

Mediante los ejemplos de este capítulo, iremos desarrollando esa rutina y, al final, podremos incorporarla a nuestro «procesador de pantallas».

Ejemplos:

Para imprimir un determinado carácter en la pantalla, partiremos de su código, es decir, entraremos en la rutina con el acumulador (registro «A») conteniendo el código del carácter. Pero el hecho de imprimir un determinado carácter, implica activar una serie de pixels que, todos juntos, formarán la letra o signo en cuestión. Cada carácter se compone de 64 pixels, es decir, de 8 octetos; para imprimirlo, tenemos que transferir esos 8 octetos desde una tabla donde estén definidos todos los caracteres, hasta la pantalla. Las tablas que contienen la definición de caracteres, se suelen denominar «Fonts». La ROM del Spectrum incluye un «Font» que contiene la definición de los 96 caracteres con códigos comprendidos entre 32 y 127 (ambos inclusive). Nosotros podríamos diseñar una tabla que contuviera nuestros propios caracteres y con el diseño que más nos gustara; pero, de momento, vamos a utilizar el Font de la ROM para no complicarnos demasiado la vida.

Cada carácter viene definido por 8 bytes consecutivos, por lo que el Font es, en realidad, una tabla de 96 elementos donde cada elemento tiene 8 bytes de longitud. El primer carácter será el de código 32 (espacio) que tiene los ocho octetos a «0». Para movernos por la tabla y apuntar al primer octeto de un determinado carácter, deberemos hace lo siguiente: primero, multiplicamos por 8 el código del carácter; a continuación, sumamos al resultado un número que es la dirección de inicio de la tabla menos 256 (tenga en cuenta que el primer código es 32 y 32*8=256. De esta forma, empieza nuestra rutina de impresión a la que hemos denominado «IMP—A» precisamente porque sirve para imprimir el carácter cuyo código esté en «A». Vayamos viendo su listado:

```
100 IMP_A  LD   DE,(CHARS)
110        LD   H,0
120        LD   L,A
130        ADD  HL,HL
140        ADD  HL,HL
150        ADD  HL,HL
160        ADD  HL,DE
170        EX   DE,HL
```

La dirección base del Font está en la variable del Sistema «CHARS» y desde ahí la leemos en la línea 100. En 110 y 120, cargamos en «HL» el código del carácter. En 130, 140 y 150 lo multiplicamos por 8. En 160 lo sumamos a la dirección base y en 170 pasamos la dirección del carácter en el Font al registro «DE».

A la salida de esta rutina, ya tenemos el registro «DE» apuntando al primero de los ocho octetos que definen el carácter.

A continuación, necesitamos hallar la dirección de pantalla correspondiente al primer octeto del lugar donde vayamos a colocar el carácter según indiquen las coordenadas. Pretendemos que esta rutina sea compatible con el Sistema operativo del Spectrum, es decir, que podamos utilizarla alternándola con sentencias «PRINT». En este caso, lo mejor es que utilicemos las coordenadas en curso del Basic.

Existe una variable del Sistema llamada «S_POSN» (Screen Position) y situada en las direcciones 23688 y 23689 que contiene, siempre, las coordenadas en curso; es decir, las de la celdilla de pantalla siguiente a la última que se haya impreso. El primer octeto contiene la columna y el segundo la línea. Para «incordiar» un poco al sufrido programador, estos valores están almacenados de una forma un tanto curiosa. En realidad, lo que se almacena no es el número de línea, sino 24 menos ese número, de forma que tendremos un «24» para la línea 0 y un «1» para la línea 23. De igual forma ocurre con el número de columna. En este caso, se almacena 33 menos el número de columna, así que tendremos «33» para la columna 0 y «2» para la columna «31». No es un problema demasiado grave y más adelante veremos cómo se resuelve.

Además de «S_POSN», existe otra variable denominada «DF_CC» que almacena la dirección, en el archivo de pantalla, del primer scan de la celdilla apuntada por «S_POSN» así que, de momento, nos ahorramos calcularla. No obstante, después de imprimir el carácter será necesario actualizar el contenido de estas variables.

Por el momento, vamos a tomar el contenido de «DF_CC» como dirección de destino de nuestro carácter. Por tanto, nuestra rutina continúa:

```
180         LD   HL,(DF_CC)
```

Ya tenemos en «DE» la dirección de nuestro carácter en el Font, y en «HL» la dirección del archivo de pantalla a partir de donde habrá que imprimirlo. Lo normal ahora, sería entrar en un bucle que fuera leyendo cada octeto apuntado por «DE» y almacenándolo donde apunta «HL» incrementando «DE» y «H» en cada pasada (recuerde que incrementamos el octeto alto de la dirección de pantalla para pasar al siguiente scan dentro del mismo carácter, por eso incrementamos «H» y no «HL»).

El bucle podría ser algo así como:

```
190 IMPR   LD   B,8
200 BUCLE  LD   A,(DE)
210        LD   (HL),A
220        INC  DE
230        INC  H
240        DJNZ BUCLE
```

A la salida del bucle, tendríamos en pantalla los ocho bytes que definen el carácter y sólo nos faltaría actualizar las coordenadas antes de retornar. Hemos llamado «IMPR» a la rutina porque sirve para imprimir; vamos a ver detenidamente cómo funciona.

En la línea 190 cargamos un «8» en «B» porque será el número de iteraciones de nuestro bucle. En 200 cargamos en «A» el primer byte del carácter y los transferimos a la pantalla en 210. En 220 y 230 incrementamos los punteros y en 240 cerramos el bucle.

Esta rutina hace lo mismo que «PRINT» en el Basic, salvo que no maneja colores ni códigos de control; en compensación, es considerablemente más rápida.

Podríamos hacer una rutina de impresión que manejara los colores (archivo de atributos) pero no tendría demasiado sentido hacer algo que ya tenemos en la ROM, así que vamos a intentar algo más original. Podemos modificar esta rutina de impresión para que nos imprima letras cursivas o negritas. La letra cursiva la obtenemos desplazando los tres primeros octetos del carácter a la derecha y los tres últimos a la izquierda, dejando los dos centrales inalterados; este tipo de letra también se denomina «itálica». La negrita podemos obtenerla haciendo un «OR» de cada byte con el mismo desplazado a la derecha; este tipo de letra también se denomina «bold». Estas dos posibilidades no son excluyentes, no hay ningún problema en que la letra sea cursiva y bold al mismo tiempo, de hecho, es la letra más bonita.

De alguna forma tiene que saber la rutina qué tipo de letra queremos; para ello utilizaremos dos «flags» que serán dos bits de la dirección de memoria 23681 (5C81h) que etiquetamos como «BAND» (por «banderas»). El primer bit por la derecha (el de menos peso) indicará letra cursiva cuando esté a «1». El segundo, indicará letra bold (también cuando esté a «1»). En el caso de que ambos estén a «0» se imprimirá letra normal y si ambos están a «1» la letra será cursiva y bold.

Los dos bits serán comprobados en cada pasada del bucle para actuar en consecuencia. También será necesario

listado la pseudo-instrucción «ORG 60000».

Ahora, vamos a hacer una pequeña rutina que ensamblaremos a partir de 60500:

```
1000 TEST  ORG  60500
1010       LD   HL,MENS
1020       LD   B,20
1030 LOOP  LD   A,(HL)
1040       PUSH BC
1050       PUSH HL
1060       CALL IMP_A
1070       POP  HL
1080       INC  HL
1090       POP  BC
1100       DJNZ LOOP
1110       RET
1120 MENS  DEFM "curso C/M "
1130       DEFM "MICROHOBBY"
```

Hay muchas cosas nuevas en esta rutina así que vamos a verla con detenimiento. En primer lugar, hemos utilizado por segunda vez el pseudo-nemónico «ORG». Aunque ensambláramos las dos rutinas juntas, no hay problema por ello. No es necesario que «ORG» vaya al principio del código fuente, además, puede utilizarse tantas veces como se quiera. Simplemente, cada vez que el ensamblador se encuentre con un «ORG», colocará el siguiente bloque de código a partir de esta dirección. De esta forma, es posible ensamblar de una sola vez varios bloques de código que ocupen direcciones distintas.

En segundo lugar, hemos vuelto a utilizar la instrucción «CALL» así que no nos quedará más remedio que explicar cómo se ensambla.

```
*HISOFT GENS3M ASSEMBLER*
        ZX SPECTRUM

 Copyright HISOFT 1983
 CURSO C/M MICROHOBBY

Pass 1 errors: 00

        10 *C-
        20 *D+
        30 ;IMPRIME_CODIGO_EN_"A"
        40 ;
60000   90         ORG     60000
60000  100 IMP_A   LD      DE,(CHARS)
60004  110         LD      H,0
60006  120         LD      L,A
60007  130         ADD     HL,HL
60008  140         ADD     HL,HL
60009  150         ADD     HL,HL
60010  160         ADD     HL,DE
60011  170         EX      DE,HL
60012  180         LD      HL,(DF_CC)
60015  190 IMPR_1  LD      B,8
60017  200 BUC_1   LD      A,(DE)
60018  210         LD      (HL),A
60019  220         LD      A,(BAND)
60022  230         AND     1
60024  240         JR      Z,NOCURS
60026  250         SRL     (HL)
60028  260         LD      A,B
60029  270         CP      5
60031  280         JR      NC,NOCURS
60033  290         SLA     (HL)
60035  300         CP      3
60037  310         JR      NC,NOCURS
60039  320         SLA     (HL)
60041  330 NOCURS  LD      A,(BAND)
60044  340         AND     2
60046  350         JR      Z,NOBOLD
60048  360         LD      A,(HL)
60049  370         SRL     A
60051  380         OR      (HL)
60052  390         LD      (HL),A
60053  400 NOBOLD  INC     DE
60054  410         INC     H
60055  420         DJNZ    BUC_1
       430 ;
60057  440         LD      DE,(S_POSN)
60061  450         LD      HL,#1821
60064  460         SBC     HL,DE
60066  470         EX      DE,HL
60067  480         INC     E
60068  490         LD      A,E
60069  500         CP      32
60071  510         JR      C,SIGUE
60073  520         LD      E,0
60075  530         INC     D
60076  540         LD      A,D
60077  550         CP      21
60079  560         JR      C,SIGUE
60081  570         CALL    SCROLL
60084  580         LD      DE,#1400
60087  590 SIGUE   PUSH    DE
60088  600 DIR     LD      A,D
60089  610         AND     #07
60091  620         RRC     A
60093  630         RRC     A
60095  640         RRC     A
60097  650         OR      E
60098  660         LD      E,A
60099  670         LD      A,D
60100  680         AND     #18
60102  690         OR      #40
```

deberá encargarse de que nunca se vuelquen más de 22 líneas de una sola vez.

En realidad, la pantalla tiene 24 líneas y podríamos haberlas utilizado todas, pero hemos querido que la rutina sea compatible con el Sistema Operativo Basic, así que hemos respetado las dos líneas inferiores para que las pueda seguir usando el canal «K». No obstante, quien desee puede modificar la rutina para que trabaje sobre toda la pantalla. Para ello, lo único que hay que hacer es poner «23» en lugar de «21» en la línea 550 y « 1600» en lugar de « 1400» en la línea 580.

Nos hemos quedado «colgados» con unos puntos suspensivos después de la etiqueta «SIGUE» donde teníamos en «DE» las nuevas coordenadas. Ahora tenemos que calcular la nueva dirección del archivo de pantalla correspondiente a estas coordenadas. Vamos a ver cómo seguiría la rutina:

```
590 SIGUE  PUSH DE
600 DIR    LD   A,D
610        AND  #07
620        RRC  A
630        RRC  A
640        RRC  A
650        OR   E
660        LD   E,A
670        LD   A,D
680        AND  #18
690        OR   #40
700        LD   D,A
```

Primero, salvamos «DE» en la pila ya que luego lo necesitaremos. Después, entramos en la rutina «DIR» que se explicó anteriormente. En esta rutina, lo que hacemos es calcular la dirección de pantalla que corresponde a las nuevas coordenadas. A la salida, tendremos esta dirección en «DE». Vamos a ver qué hacemos con ella:

```
720        LD   (DF_CC),DE
730        POP  DE
740        LD   HL,#1821
750        SBC  HL,DE
760        LD   (S_POSN),HL
770        RET
```

Primero guardamos en «DF-CC» la nueva dirección de pantalla, luego recuperamos las coordenadas de la pila, las invertimos restándolas de 1821h y almacenamos el resultado en «S-POSN»; finalmente, retornamos en la línea 770.

Para que la rutina «IMP-A» esté completa, sólo nos falta definir el valor de algunas etiquetas, así que vamos a ello:

```
780 CHARS  EQU 23606
790 DF_CC  EQU 23684
800 BAND   EQU 23681
810 S_POSN EQU 23688
820 SCROLL EQU #0DFE
```

Con esto termina la rutina «IMP-A». Cada vez que la llamemos con el código de un carácter en «A», nos imprimirá ese carácter y actualizará las variables del sistema necesarias para que el Basic siga funcionando.

Hasta ahora, todas las rutinas que hemos hecho podían ser llamadas directamente desde el Basic. Sin embargo, no es este el caso de «IMP-A» ya que, si la llamamos con USR, no sabremos el contenido de «A» y nos imprimirá un carácter aletorio. En realidad, esta rutina no ha sido concebida para llamarla con USR, aunque luego veremos una forma de usarla directamente desde Basic.

Por si algún lector aún no se había dado cuenta, lo que estamos intentando es construir todo un programa para gestionar la pantalla. Hemos hecho una serie de rutinas que trabajaban de forma distinta según el valor que contuviera el registro «A» (Borrado por trozos, intercambio de bloques, etc.). Para todas ellas, el valor de «A» tenía que ser menor de «32». Pues bien, esta rutina será, en su día, una subrutina del programa para gestionar la pantalla, concretamente, se encargará de imprimir un carácter cuando el contenido de «A» sea mayor de 32.

De momento, no nos parece bien hacer esperar al lector, así que vamos a ver de qué forma podemos utilizar esta rutina. En principio, lo más sencillo es coger una zona de memoria y almacenar en ella los códigos de los caracteres que componen un determinado mensaje para, luego, hacer un bucle que fuera cargando los códigos en «A» uno por uno y llamando a esta rutina para que los imprimiera.

Vamos a imprimir el mensaje: «curso C/M MICROHOBBY» utilizando esta rutina. «IMP-A» es reubicable, pero vamos a ensamblarla a partir de la dirección 60000. Para ello, añadimos al principio del

comprobar en qué scan del carácter nos encontramos para saber hacia dónde hay que desplazar el byte cuando imprimamos en cursiva. En lugar de desplazar los tres primeros scans a la derecha y los tres segundos a la izquierda, vamos a seguir un método que resulta más fácil de programar: Cuando estemos imprimiendo en cursiva, primero desplazamos el byte a la derecha y si estamos en uno de los tres primeros scans lo dejamos así y saltamos al final; luego lo desplazamos un lugar a la izquierda y si estamos en uno de los 5 primeros scans lo dejamos así y saltamos al final; finalmente, desplazamos el byte otro lugar a la izquierda. De esta forma, conseguimos el efecto de inclinación deseado. Otra consideración a tener en cuenta es procesar primero el efecto de cursiva y luego el de bold, ya que si lo hiciéramos al revés, cuando utilizáramos ambos efectos no obtendríamos el resultado deseado.

A esta rutina que puede imprimir en bold y cursiva la llamaremos «IMPR_1» y deberá ir en lugar de la «IMPR» que vimos anteriormente. Vamos a ver el listado:

```
190 IMPR_1 LD   B,8
200 BUC_1  LD   A,(DE)
210        LD   (HL),A
220        LD   A,(BAND)
230        AND  1
240        JR   Z,NOCURS
250        SRL  (HL)
260        LD   A,B
270        CP   5
280        JR   NC,NOCURS
290        SLA  (HL)
300        CP   3
310        JR   NC,NOCURS
320        SLA  (HL)
330 NOCURS LD   A,(BAND)
340        AND  2
350        JR   Z,NOBOLD
360        LD   A,(HL)
370        SRL  A
380        OR   (HL)
390        LD   (HL),A
400 NOBOLD INC  DE
410        INC  H
420        DJNZ BUC_1
```

Si ambos flags están a «0» saltaremos primero a «NOCURS» y luego a «NOBOLD» con lo que la rutina será igual que «IMPR».

Supongamos que el primer bit de «BAND», es decir, el flag de cursiva, está a «1». En ese caso, saldremos de la línea 230 con el indicador de «cero» a «0» y no se producirá el salto a «NOCURS» sino que el programa seguirá por la línea 250.

En esta línea, empezamos por desplazar a la derecha el octeto que acabábamos de transferir al archivo de pantalla. A continuación, en las líneas 260, 270 y 280, comprobamos si nos hallamos en uno de los primeros tres scans del carácter, en cuyo caso, saltamos a «NOCURS». Esta comprobación se lleva a cabo mirando si el contenido de «B» (contador del bucle) es mayor de «5», en cuyo caso, al comparar con «5» no se producirá acarreo.

Si no es así, continuamos en la línea 290 donde desplazamos el octeto a la izquierda. Seguimos teniendo en «A» el contenido de «B», así que lo comparamos con «3» para ver si estamos en los dos scans de la mitad del carácter. Si es así, saltamos a «NOCURS». Si no, ejecutamos un desplazamiento más a la izquierda en la línea 320.

De esta forma, los tres primeros scans quedán desplazados a la derecha, los dos de en medio quedan tal como están y los tres últimos quedan desplazados a la izquierda lográndose el efecto de letra cursiva. Ver Figura 9-17.

En las líneas 330, 340 y 350 comprobamos el flag de «bold» (letra negrita). Supongamos que está a «1» en cuyo caso no se produce el salto a «NOBOLD» y se continúa en la línea 360. En esta línea, cargamos en «A» el octeto que acabamos de transferir. En 370 lo desplazamos a la derecha. En 380 le hacemos un «OR» con él mismo y, finalmente (en la línea 390), colocamos el resultado en el lugar correspondiente del archivo de pantalla.

A partir de «NOBOLD», la rutina continúa de forma normal: se incrementan los punteros «DE» y «H» y se cierra el bucle para el siguiente scan o se sale del mismo si ya se han completado los ocho scans correspondientes a un carácter.

Ya tenemos el carácter impreso en la pantalla, ahora nos queda actualizar las variables del Sistema que contienen las coordenadas y la dirección en el archivo de presentación visual.

El procedimiento a seguir es el siguiente:

1. Se leen las coordenadas antiguas.
2. Se incrementa el número de columna.

3. Si es mayor de 31, se pone a cero y se incrementa el número de línea.

4. Si ésta es mayor de 21, se hace «scroll» hacia arriba de una línea y se pone 21 como número de línea.

5. Se calcula la nueva dirección de pantalla.

6. Se almacena la nueva dirección de pantalla.

7. Se almacenan las nuevas coordenadas.

Como indicamos antes, las coordenadas se encuentran en una variable del Sistema denominada «S-POSN» cuya dirección es 23688 y están almacenadas de forma invertida, es decir, no tenemos el número de líneas o columnas que van escritas, sino las que nos faltan para llegar al final más dos. Por tanto, para recuperar las coordenadas correctas tal y como nos interesan deberemos restar de 1821h el contenido de la variable «S-POSN». Vayamos viendo el listado:

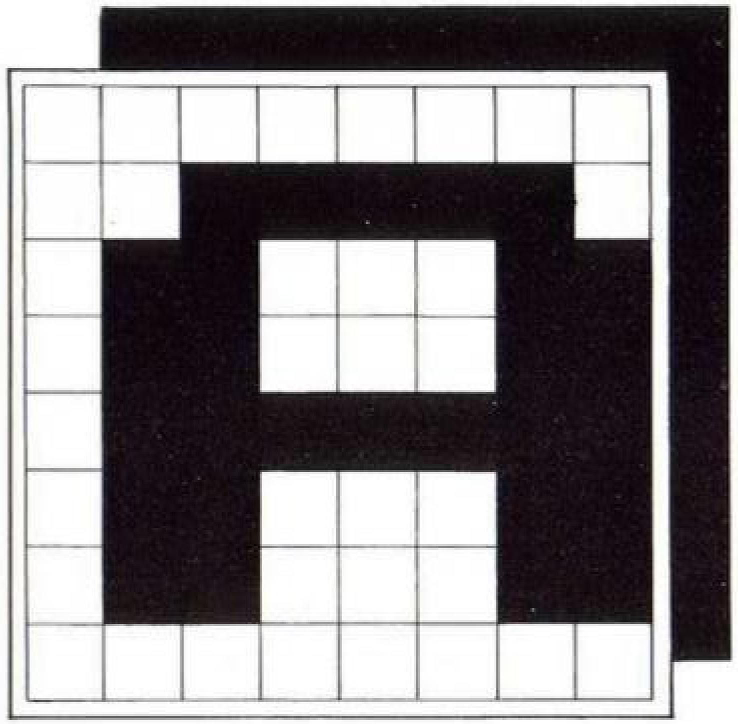


Fig. 9-17a. Formato de la letra «A».

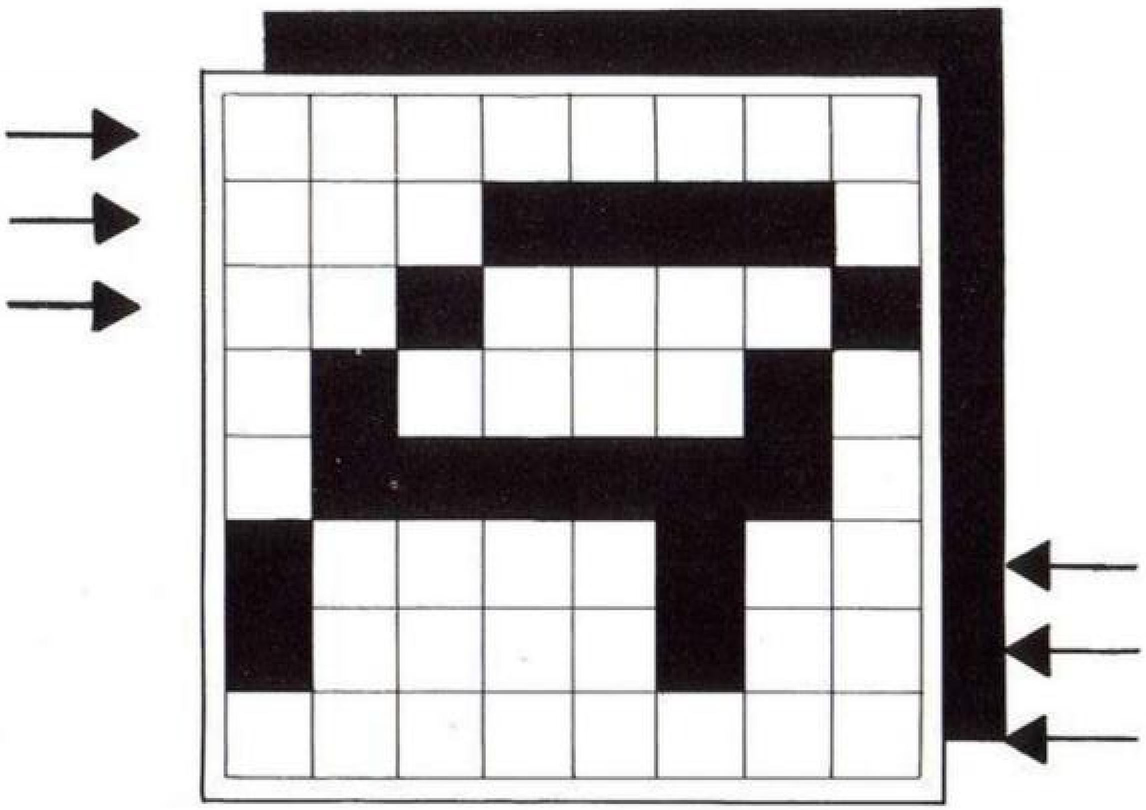


Fig. 9-17b. Formato de la «A» en cursiva.

```
440        LD   DE,(S_POSN)
450        LD   HL,#1821
460        SBC  HL,DE
470        EX   DE,HL
```

Empezamos por leer en «DE» el contenido de «S-POSN», luego lo restamos de 1821h y transferimos el resultado, de nuevo, a «DE». Por tanto, a la salida de esta rutina tenemos en «D» la línea donde hemos impreso nuestro carácter y en «E» la columna. Ahora, empezaremos por incrementar el número de columna. Sigamos con el listado:

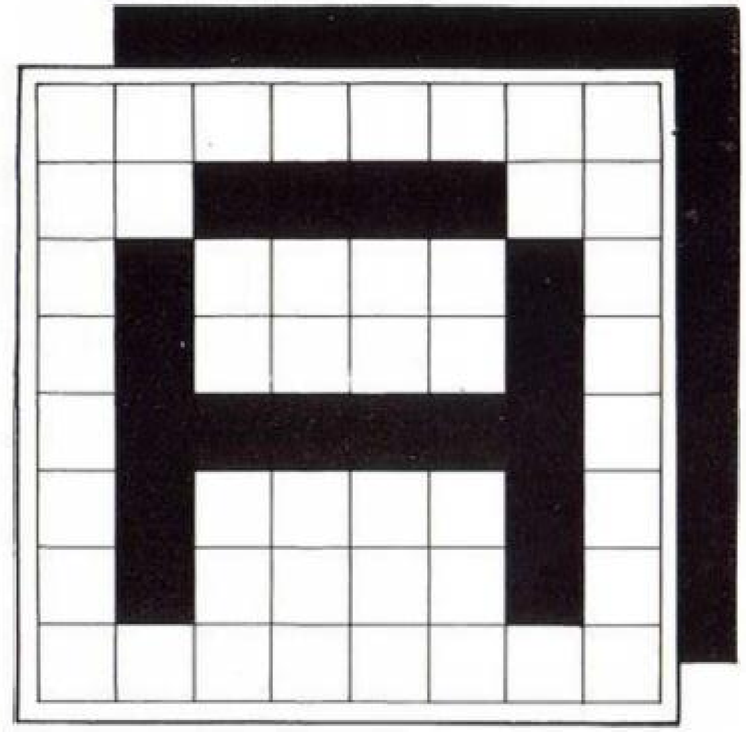


Fig. 9-17c. Formato de la «A» en bold o negrita.

```
480        INC  E
490        LD   A,E
500        CP   32
510        JR   C,SIGUE
520        LD   E,0
530        INC  D
540        LD   A,D
550        CP   21
560        JR   C,SIGUE
570        CALL SCROLL
580        LD   DE,#1400
590 SIGUE  ......
```

De momento, ignore la línea 570. Luego veremos para qué vale. Empezamos por incrementar «E». En las líneas 490, 500 y 510 comprobamos si es menor de 32 en cuyo caso, ya estarían actualizadas las coordenadas y saltaríamos a la etiqueta «SIGUE». Si el valor de «E» después de incrementado es 32 o mayor (nunca puede ser mayor de 32, pero el mismo trabajo nos cuesta comprobar si es igual o mayor), continuaríamos en la línea 520 donde cargamos un «0» en «E» e incrementamos «D» para colocarnos al principio de la siguiente línea.

Ahora debemos comprobar si hemos alcanzado la línea 21. Si no es así, saltamos a «SIGUE». Si, por el contrario, hemos llegado al final de la pantalla, será necesario realizar un «scroll» hacia arriba, es decir, subir toda la pantalla una línea y colocar la nueva posición de impresión al principio de la línea 20, lo que conseguimos cargando 1400h en «DE».

Para realizar el «scroll» hacia arriba, podíamos haber escrito una rutina en C/M que lo hiciera, pero como no nos gusta trabajar de balde y ya tenemos esa rutina en la ROM, hemos decidido que lo mejor es utilizarla. La instrucción «CALL» de la línea 570 es una llamada a subrutina. Aún no hemos estudiado estas instrucciones por lo que, de momento, nos conformaremos con saber que «CALL SCROLL» nos sirve para subir una línea toda la pantalla. Por supuesto, en capítulos posteriores estudiaremos no sólo las instrucciones de llamada a subrutinas, sino también algunas subrutinas de la ROM que, como «SCROLL», pueden resultarnos muy útiles al programar.

Como es evidente, cuando utilicemos esta rutina para imprimir, nunca nos saldrá el famoso mensaje «Scroll?» («Sigo?» en los españoles). De hecho, el Spectrum es el único ordenador donde ocurren estas cosas; lo normal es que la pantalla suba sin esperar a que lo digamos. El programa que utilice esta rutina para el volcado de datos por pantalla,

Cómo entrar en un programa y averiguar sus secretos

LA BIBLIA DEL «HACKER» (III)

José Manuel LAZO

En el listado de un programa Basic pueden hacerse determinadas alteraciones de tal forma que sea imposible averiguar su contenido e incluso que, al intentarlo, el propio listado se modifique creando una gran confusión.

Un caso típico es que cuando hacemos LIST, no sale nada, y además se nos presenta el cursor con una interrogación. Esto es porque se ha utilizado un control de AT con coordenadas falsas. Todo esto es en realidad basura dentro del listado para evitar que se vea. Pero la mejor manera de aprender todo esto es practicando, por lo que vamos a exponer unos ejemplos sencillos:

Primero teclea lo siguiente teniendo la memoria del ordenador limpia:

```
   1 REM (pon aqui tres espacios
 en blanco)
   2 REM (otros cuantos espacios
)
1000 INPUT "Pokes? ";n
1010 LET direccion=(PEEK 23635+2
56*PEEK 23636)+5
1020 FOR a=direccion TO direccio
n+n
1030 INPUT "Valor? ";b: POKE a,b
1040 NEXT a
1050 STOP
```

Teclea GOTO 1000 y prueba algunos controles: Primero uno, por lo que responde a la primera pregunta con un 1, y a la segunda con un 6. Si haces LIST verás que el texto de la primera línea se ha desplazado a la columna central de TAB, tal y como si hubiéramos utilizado PRINT con coma. Este control es el 6.

Responde ahora a la primera pregunta con 1, y a la segunda con 22; este es el control de AT. Puesto que los dos siguientes valores son dos 32, que corresponden al espacio, cuando pulsemos «ENTER» para hacer un listado automático no nos saldrá y tendremos el cursor junto con una interrogación. Sin embargo, al dar la orden LIST saldrá inmediatamente el error «entero fuera de rango».

Si tecleamos GOTO 1000 e introducimos el control de AT con unos valores adecuados cambiaremos las coordenadas del listado. Responde a la primera pregunta con 3 y a las tres siguientes con 22, 10 y 10. El listado aparecerá dividido en dos trozos.

Por último, vamos a ver la forma de sobreescribir en el listado: responde a la primera pregunta con 3, y a las 3 siguientes con 22, 0 y 0. Veremos cómo el número de la primera línea ha desaparecido imprimiéndose el texto de la primera línea REM en las coordenadas 0,0.

Otra consecuencia de tener basura en el listado es que si conseguimos editar la línea, no la podemos modificar debido a que constantemente está sonando el zumbador de alarma por el error que, intencionadamente, se ha introducido en ella.

Esto último también puede ser debido a que en el listado exista un CLEAR que sitúe el RAMTOP excesivamente bajo para permitir la edición.

La basura de un listado se introduce con la finalidad de corromper el programa si tratamos de editar líneas o modificarlo en alguna de sus partes.

Hay que buscar alguna forma de poder ver un listado sin tener que modificar ninguna línea.

En primer lugar, es conveniente saber algunas cosas acerca de cómo se organiza un programa Basic en la memoria. Las líneas de programa se guardan en la memoria de la siguiente forma: primero dos octetos que indican el número de línea de que se trata. Si nosotros pokeamos en esa dirección con otro valor, cambiaremos el número de línea. Podemos ponerlo a «0» e incluso a un número imposible, mayor de 9999, dado que en dos octetos cabe cualquier número menor de 65535. Obviamente el efecto contrario también es posible, es decir, podemos cambiar el número para que sea legal.

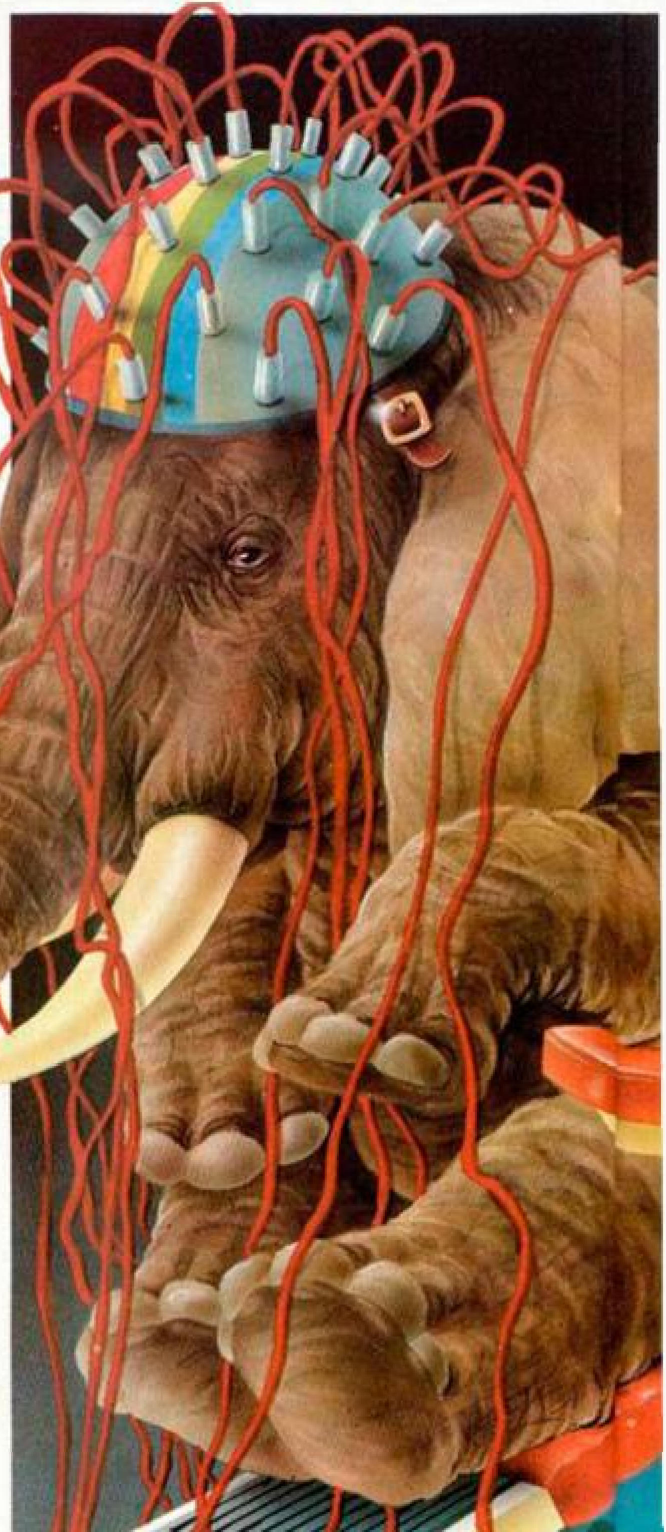
Estos dos octetos se ponen al revés de como sería de esperar, el primero es el más significativo, y el segundo el de menor peso, esto es así para que el intérprete funcione más rápido.

Después de estos dos bytes vienen otros dos que indican la longitud de la línea incluyendo el código de «Enter» del final. Seguro que ya se te está ocurriendo que podemos variar también esta información para complicar más las cosas. Ello es posible haciendo que estos octetos contengan unos datos falsos, marcando más o menos longitud de lo normal. Lo hemos visto en muy pocos programas dado que también confunde al SO, y una cosa hay que tener muy clara, todas las protecciones a nivel Basic que podemos encontrar tienen la particularidad de que confunden el listado, pero nunca al SO.

En el texto de la línea se guardan todos los tokens y literales por sus respectivos códigos ASCII, pero hay una particularidad: los números. Después del texto de la línea viene un control de «Enter» (13), que marca la frontera entre líneas.

CHEQUEO DE LA MEMORIA EN EL SPECTRUM (I)

Miguel SEPULVEDA y Adolfo PEREZ



En alguna ocasión, cuando un determinado programa se niega a funcionar, nos ha asaltado la terrible duda: «¿Estará estropeado mi ordenador?» La verdad es que es bastante difícil precisar esto, sobre todo cuando no es del todo evidente. En la memoria del Spectrum puede haber posiciones estropeadas y pasar desapercibido hasta que usemos un programa que las necesite. ¿Cómo podemos averiguar si la memoria se encuentra en perfectas condiciones?

El programa que presentamos en esta ocasión nos permitirá probar la memoria del ordenador y mostrarnos, de forma gráfica, los posibles errores. La prueba consistirá en rellenar la memoria con una serie de valores para después leerlos y compararlos. Si lo leído no coincide con lo escrito, detectaremos los bits en que se producen los fallos y pondremos en pantalla, en forma intermitente, los circuitos integrados que es probable que se encuentren estropeados dentro del esquema general de Spectrum.

En los 2 programas que integran esta rutina, una primera función correspondería a la prueba de memoria propiamente dicha, y la segunda a la creación de una pantalla/esquema conteniendo todas las pastillas, conectores..., etc., que componen el Spectrum y sobre la que mostraremos en modo flash las pastillas en fallo.

El test de memoria

El software correspondiente a esta prueba, consta de dos partes, una parte Basic (programa 1), y otra en código máquina (listado C.M.).

Una vez cargado y ejecutado el programa, se nos pide la configuración de nuestro Spectrum, esto es, si es de 16 K o de 48 K. Esto se hace para calcular la última posición de memoria que se debe comprobar y que será 32767 para el de 16 K o 65535 para el de 48 K. A continuación, se nos pide la posición donde queremos cargar la rutina en código máquina que nos comprobará la memoria. La dirección más baja en que podremos cargarla, será la 25000. La más alta dependerá de la configuración del micro. Se ha fijado 32767—500 = 32267 para el de 16 K y 65035 para el de 48 K.

Una vez introducidos los datos anteriores, pondremos el cassette en marcha para cargar la rutina MEMORY (que previamente habremos grabado en cinta), en la dirección deseada. Esta rutina será la encargada de realizar las pruebas.

Si observamos el listado del programa 1, vemos que después de cargar la rutina, se hace un POKE en la dirección 16384. El valor introducido en esa posición será un indicativo para la rutina MEMORY que después veremos con detalle. A continuación del POKE se hace RANDOMIZE n y como consecuencia de él, en la variable SEED queda almacenada la última dirección de memoria que tendremos que probar. Luego, se arranca la rutina por la dirección de carga elegida.

La prueba consistirá en la escritura y



ra no destruirlo. La forma en que vamos a hacerlo será la siguiente:

Inicialmente, cargaremos la rutina en la dirección de memoria deseada, delante de la rutina reservaremos 20 posiciones que utilizaremos como pila de stack propia. Luego, pasaremos la prueba desde la dirección primera de RAM, hasta la dirección en que se encuentre el stack. Si la prueba es correcta, desplazaremos la rutina hasta una posición dentro de la zona probada de forma que el final de la rutina quede dentro de dicha zona. A continuación, la prueba se repite desde el final de la rutina hasta el final de la memoria. Gráficamente podemos ver esto en la figura 3.

En dicha figura, si la rutina ha sido cargada inicialmente en la posición 25000, en la dirección 24980 colocamos el stack, y entonces, en la primera parte de la prueba, probaremos desde la posición 16384 hasta la 24979. Vamos a suponer que la prueba es correcta, en cuyo caso, desplazaremos el programa hasta una dirección fija X, de manera que el programa está ahora dentro de la zona probada inicialmente. La segunda parte consistirá en probar desde el final del programa hasta el final de la memoria. Como observamos en la figura 3, hay un trozo de memoria que se prueba dos veces, lo cual no es ningún problema. Vamos a considerar el caso en que en la primera parte de la prueba se hubiera producido un error. En este caso habríamos anotado, a partir de la dirección X, en qué dirección se produjo el error. Todos los posibles errores los anotaríamos a continuación. El programa lo desplazaríamos a partir de la última dirección con error.

La dirección X es una dirección fija y coincide con el inicio del buffer de la impresora, esto es, la 23296.

Durante el transcurso de la prueba, lectura en toda la memoria de cuatro valores consecutivamente, de forma que cada uno de ellos es escrito, leído y comprobado. Como en todo momento el programa reside en memoria, tendremos que utilizar cualquier artimaña pa-

PROGRAMA 1

```
  10 CLEAR 24999
  20 PRINT FLASH 1;AT 10,8;"PARA
 EL CASETTE"
  30 PAUSE 200
  40 INPUT "Num. de Kbytes de RA
M (16 o 48) ";n: IF n<>16 AND n<
>48 THEN GO TO 40
  50 LET n=n*1024+16383
  60 INPUT "Direccion donde quie
re cargar larutina (min. 25000,
max. ";(n-500);") ";d
  65 IF d<25000 OR d>n-500 THEN
GO TO 60
  70 PRINT FLASH 1;AT 10,3;"PON
EL CASETTE EN MARCHA"
  80 LOAD ""CODE d
  85 PRINT FLASH 1;AT 10,1;"PULS
A UNA TECLA": PAUSE 0
  90 POKE 16384,1: RANDOMIZE n:
RANDOMIZE USR d
```

LISTADO CÓDIGO MAQUINA DEL COMPROBADOR DE MEMORIA

LINEA	DATOS	CONTROL
1	CD52003B3BDDE3DD2BDD	1338
2	2BDD2BDD2BDDF9DD233A	1355
3	0040A7282FF3D9010000	779
4	ED5B765C21005BD93EA5	1106
5	CD520018793EFFCD5200	1036
6	1872D9E5D9E101140009	1056
7	E501260 1DDE5D1EBEDB0	1576
8	E1E93EA5CD520018683E	1162
9	FFCD5200186 1D9C5D93E	1356
10	8732005BDDE5DD210040	1041
11	110018 3EFF37CD5605DD	933
12	E1C178B12829DDE5D121	1488
13	060119780610CB391F30	513
14	16E55E2356212000EBF5	1011
15	C506033E80B6771910F9	987
16	C1F1E1232310E1FBCD54	1510
17	1FD2000018F73B3B1114	667
18	40DDE5E1A7ED52444D1E	1400
19	0018143B3BDDE5E11126	892
20	0119E5D9D5D9E1D1A7ED	1740
21	524D44E32323E3C5D5D5	1374
22	626B1377C50BEDB0C1E1	1382
23	5F7BBE2013230B79D3FE	1091
24	B020F47BA7C8FE5AC82F	1533
25	D1C118DDE57EABCB7CD9	1717
26	2804B14F1802B047E1D9	1015
27	18DB105A0E5A0C5A0A5A	655
28	085A065A045A025AFA59	719
29	F859F659F4597A597859	1425
30	76597459000000000000	412

INSTRUCCIONES PARA INTRODUCIR EL LISTADO DEL CODIGO MAQUINA.

1. Cargar el Cargador Universal de Código Máquina.
2. Elegir la opción INPUT.
3. Teclear el listado.
4. Una vez hayamos terminado de introducir el listado realizar un DUMP indicando como dirección 40000.
5. Para guardar el código máquina generado, indica como dirección 40000 y 294 como número de bytes.

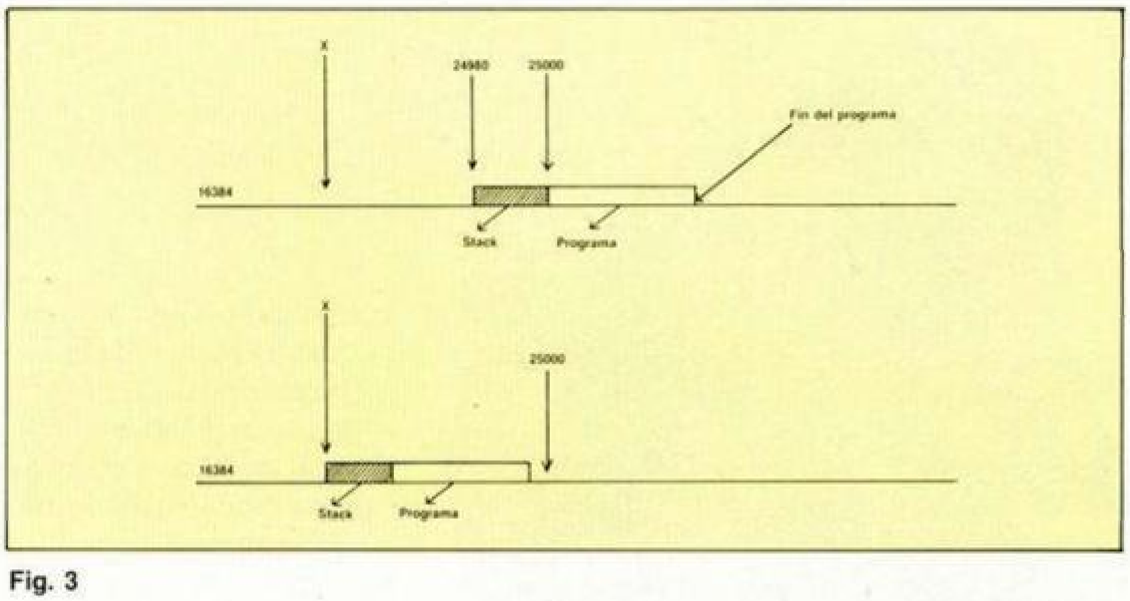
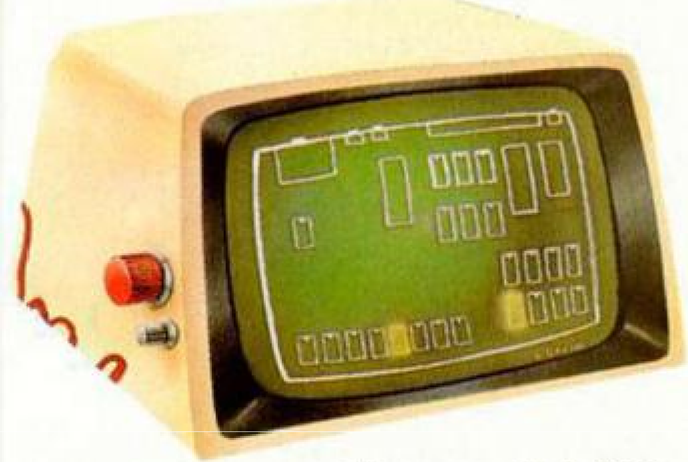


Fig. 3



aparece en pantalla una mezcla de BORDER'S y de sonidos. Una vez haya terminado, aparecerá en el ángulo superior izquierdo de la pantalla, un cuadrado en intermitente que nos indicará que pongamos el cassette en marcha para cargar la pantalla que previamente habremos salvado en cinta y que contiene el dibujo de las pastillas. Si en la prueba se ha producido algún error, la pastilla correspondiente a dicho error se pondrá en modo intermitente. La forma de salir del programa será haciendo un BREAK cuando el programa nos lo diga.

Cómo funciona la rutina

Pasemos ahora a describir la rutina que ejecuta todo esto. En la figura 4, tenemos un organigrama a muy alto nivel, de la rutina MEMORY.

En primer lugar se coloca el stack delante del programa. El stack ocupa 20 posiciones. A continuación, se pregunta si el programa ha sido ya desplazado o lo que es lo mismo, si el programa se ha ejecutado una vez ya. Anteriormente, habíamos dicho que el programa Basic que cargaba esta rutina, hacía un POKE en la dirección 16384 colocando un 1. Este valor va a servir para saber si el programa ha sido desplazado o no, puesto que cuando termine la primera parte de la prueba borraremos el 1, de forma que en la siguiente entrada, cuando se pregunte por el contenido de esa posición ya no estará a 1 y, por tanto, se tomará como segunda entrada del programa.

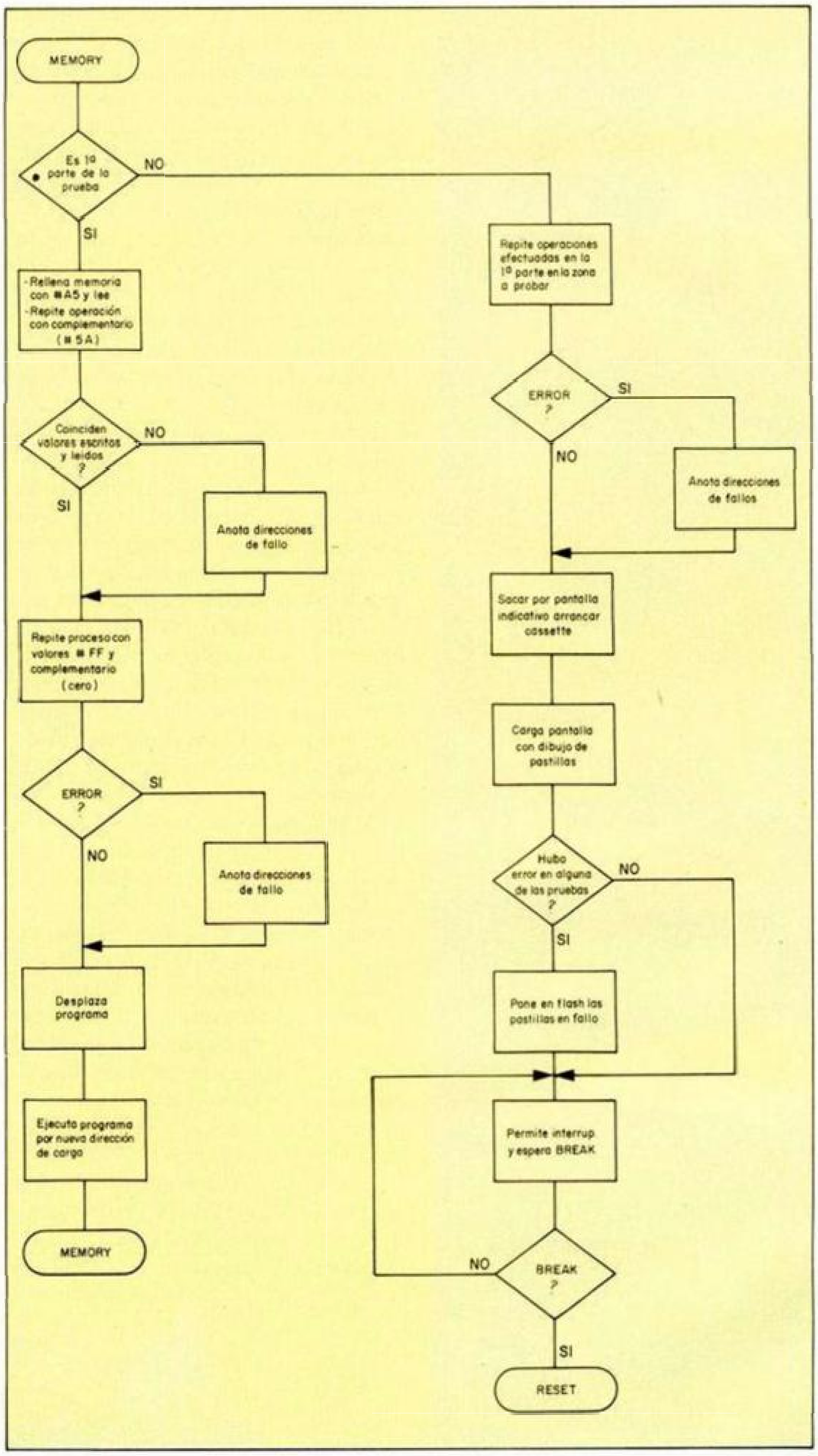
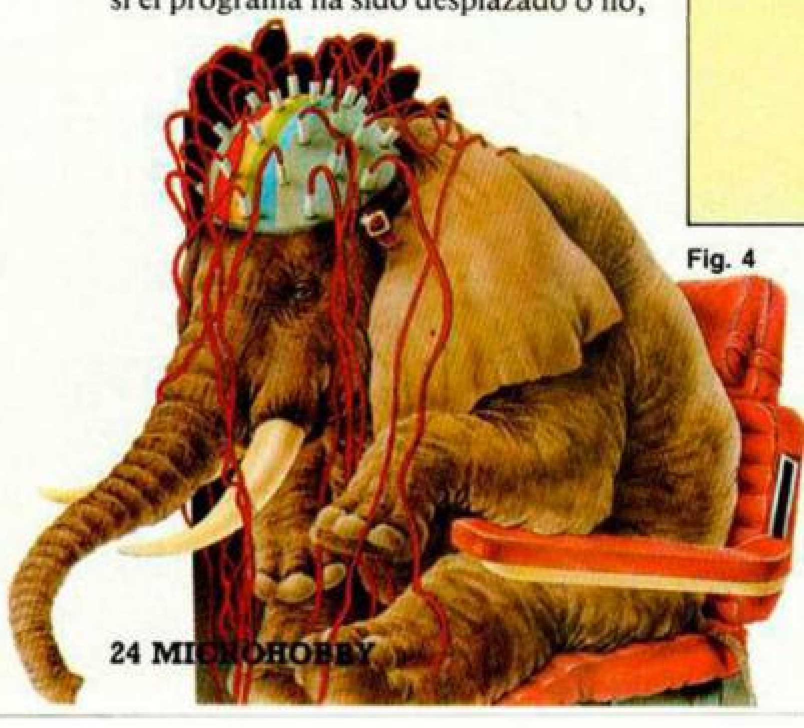
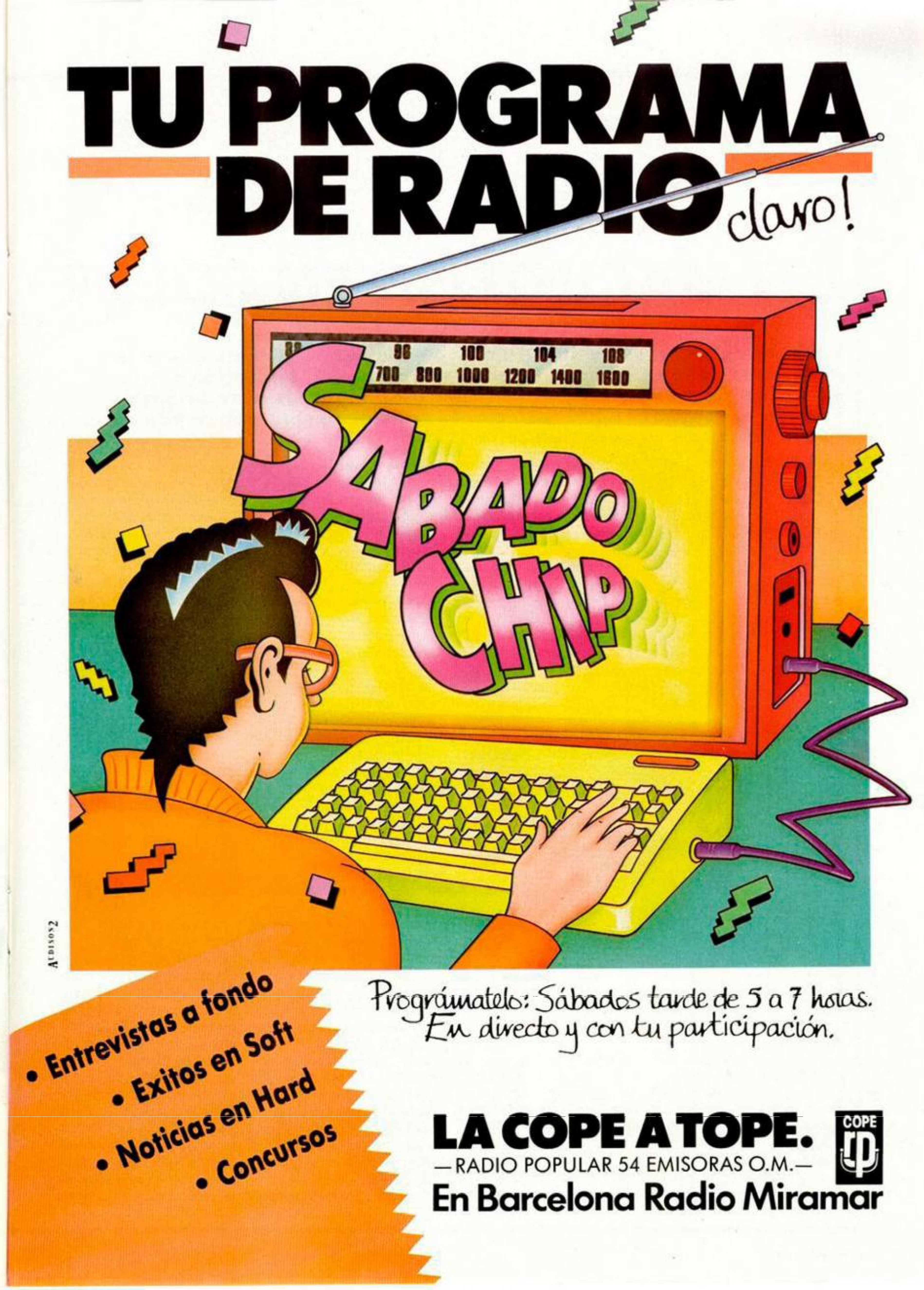


Fig. 4

La explicación de cada uno de los pasos que se llevan a cabo, el desensamble de la rutina y el programa que genera la pantalla (absolutamente necesario para el correcto funcionamiento de todo el conjunto), lo encontraréis en el próximo número.




TU PROGRAMA
DE RADIO
claro!
SABADO CHIP
96 100 104 108
700 800 1000 1200 1400 1600
• Entrevistas a fondo
• Exitos en Soft
• Noticias en Hard
• Concursos
Prográmatelo: Sábados tarde de 5 a 7 horas.
En directo y con tu participación.
LA COPE A TOPE.
—RADIO POPULAR 54 EMISORAS O.M.—
En Barcelona Radio Miramar
COPE

ART STUDIO * Utilidades * OCP, Rainbird Software

UN MACINTOSH EN COLOR

Por: Peter A. Goras

Para los detractores de Spectrum, y para quienes afirman que este ordenador es incapaz de soportar un software de aplicación «serio», este programa viene a demostrar la falsedad de sus afirmaciones. Cuando lo cargues en tu Spectrum, te parecerá estar trabajando con un Apple Macintosh... pero ¡¡en color!!

Quienes hayan tenido, alguna vez, la inolvidable experiencia de manejar un ordenador Apple Macintosh, tal vez hayan pensado que jamás habría un software para Spectrum tan cómodo, potente y fácil de utilizar como el que existe para este ordenador. Reconocemos que nosotros también lo creímos así. No obstante, nada más lejos de la realidad. Los programadores de OCP nos han demostrado, una vez más, que lo importante en un ordenador es el software.

ART STUDIO es, básicamente, un programa de diseño gráfico, es decir, sirve para crear pantallas. Dicho así, podría parecer que es uno más de los muchos que existen para Spectrum. No obstante, la particularidad de este programa es que trabaja exactamente igual que el «McPaint» de Apple; con la diferencia, eso sí, de la menor resolución gráfica del Spectrum. En compensación, ART STUDIO permite trabajar con colores.

Por supuesto, el programa es perfectamente compatible con joystick, teclado e incluso, con ¡¡RATON!! Se presenta fuertemente protegido con el sistema LENSLOCK que comentábamos en el número 67, sección Micropanorama; por lo que recomendamos a los inevitables «piratas» que, simplemente, se abstengan de intentarlo.



Con la máxima ampliación se puede retocar «al pixell» cualquier detalle.

Cuando arranca, pregunta acerca de la configuración que se va a utilizar (joystick, teclado, ratón, tipo de impresora, etc.) y se «instala» automáticamente para trabajar con esa configuración. A continuación, permite sacar una copia de seguridad del programa ya instalado, para que el usuario no tenga que hacerlo de cada vez. Finalmente, salta al sistema Lenslock de protección y pide la clave. Si ésta es correcta, comienza el programa propiamente dicho; si, por tres veces, se da una clave errónea, se borra la memoria.

Como es normal, viene acompañado de un manual de instrucciones; no obstante, la sencillez de manejo es tal que, en media hora, se puede dominar totalmente sin haber leído una sola línea del manual.

ART STUDIO está construido con una filosofía de funcionamiento que lo hace fácil de utilizar para el usuario que lo maneja por primera vez, y evita tener que memorizar complicadas secuencias de comandos. Toda la información está en pantalla, y se maneja mediante un puntero (normalmente, una flecha) que se puede desplazar por la pantalla para apuntar a una opción determinada. Para cada opción, aparece una ventana con un submenú que, de nuevo, se selecciona con el puntero. Cada modo de actuación del programa está representado por un «icono», un pequeño gráfico que nos recuerda lo que estamos haciendo. Si dibujamos, veremos un lápiz; si ampliamos, una lupa; si pintamos, una brocha o un rodillo; si utilizamos un «spray», lo veremos en la pantalla.

Existen cinco funciones básicas: arriba, abajo, derecha, izquierda y «selección». Si se utiliza joystick, serían las cuatro direcciones más el «disparo»; si se utiliza un «ratón», lo mismo; y si se maneja con teclado, una tecla, previamente seleccionada, realizará cada una de estas funciones.

En principio, aparece una flecha en el centro de la pantalla y un menú de opciones en la parte superior. No tendremos más que mover la flecha para apuntar a la opción requerida y apretar «selección». En ese momento, se «despliega» un submenú donde seleccionar, de nuevo, una subopción de la misma forma que antes. Por ejemplo: supongamos que queremos dibujar; primero seleccionamos «Paint». Aparece, entonces, un submenú que nos permite elegir «lápiz», «brocha» o

«spray». Una vez seleccionado el instrumento de dibujo, aparece un tercer menú para elegir el tamaño y la forma del trazo o la densidad del «spray». Con esto, termina la selección. Si hemos elegido un lápiz, veremos en pantalla, precisamente, un lápiz. Si hemos elegido la brocha, veremos una brocha. Y si se trata del «spray», veremos, como era de esperar, un «Spray».

Ya podemos empezar a pintar. Si mantenemos «selección» apretado al tiempo que nos movemos, el instrumento que hayamos elegido, irá dejando su rastro por la pantalla. Si soltamos «selección», simplemente se moverá de sitio sin pintar.



El uso de ventanas permite mezclar distintas imágenes.

Podemos rellenar de color una figura cerrada (Fill). Para ello, nos aparece un menú donde seleccionaremos la «trama» y, luego, un pequeño rodillo de pintor que colocaremos encima de la zona a rellenar antes de pulsar «selección».

Si queremos ampliar parte del dibujo para verlo mejor, nada más fácil: seleccionamos «magnificación» ×2, ×**4** ó ×8 y nos aparece una «lupa» en pantalla. Colocamos la lupa sobre la zona a ampliar y pulsamos «selección». Ahora podemos controlar, independientemente, cada pixel de la zona ampliada, poniéndolo a «0», a «1» o invirtiendo su valor.

Si queremos insertar un texto, seleccionamos «Text» y elegimos: de izquierda a derecha, de arriba a abajo, con letra «bold» (negrita), de lado, doble alto, doble ancho, etc. Nos aparece un cursor con forma de «I». Lo colocamos donde deseemos escribir y pulsamos «selección». Cuando acabemos, volvemos a pulsar «selección» y podemos trasladar el cursor a otro sitio o elegir otra opción del menú. Normalmente, el programa utiliza el juego de caracteres de la ROM, pero incluye un potente editor de caracteres para que podamos crear o cargar los nuestros.

También podemos crear nuestras propias tramas de relleno o darle la forma y tamaño que queramos a la «brocha».

Una de las cosas más impresionantes que se puede hacer con este programa es el manejo de ventanas. Seleccionamos «windows», definimos una ventana, volvemos a «windows» y ya podemos hacer con esa ventana lo que queramos: borrarla, invertirla, trasladarla de sitio, copiarla en otro lugar, rotarla o re-escalarla. Esto último consiste en copiarla en otro lugar y a otro tamaño, con lo que, también, variará de tamaño lo que haya en el interior de la ventana.

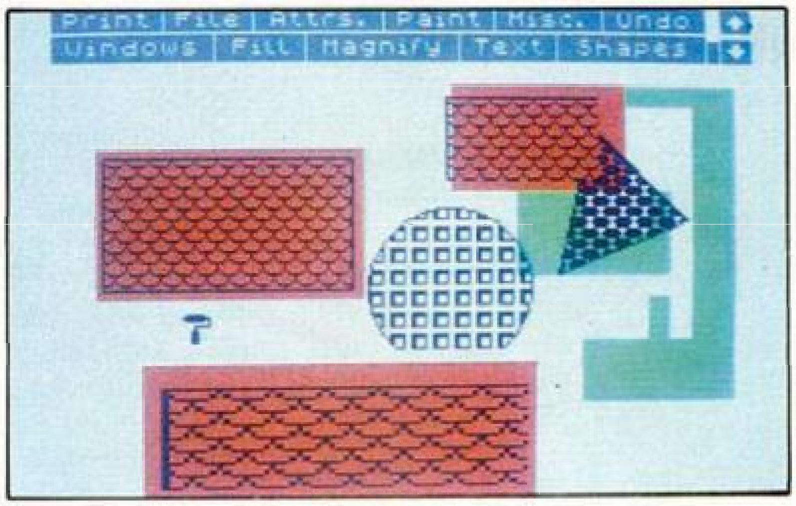
Podemos almacenar la pantalla o cargar o salvar en cinta un juego de caracteres que hayamos creado con el editor, así como convertir en caracteres una parte de la pantalla (muy útil para hacer figuras grandes a base de UDGs).



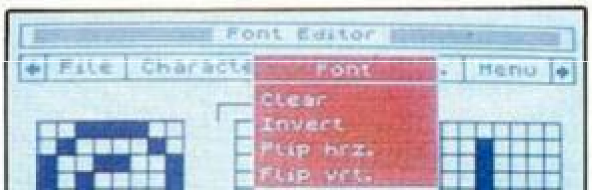
El único defecto que puede encontrarse en el ART es que sólo permite trabajar con cassette.



Todos los menús aparecen sobre la pantalla en el momento oportuno.



La opción FILL permite rellenar cualquier figura con tramas predefinidas o creadas por el usuario.



Un potente editor de caracteres permite redefinir todas las letras y UDG.

Hasta aquí, hemos explicado por encima el manejo del programa. No obstante, ART STUDIO incluye muchas más posibilidades que no hemos podido comentar, ya que llevaría un libro entero analizarlas todas. Dentro de poco, estará disponible en España. Estamos seguros de que quien lo maneje una vez, no podrá vivir sin él. La mejor definición que se puede dar de este programa es que «enamora» al usuario. Hasta que no lo cargues, no sabrás de lo que es capaz tu Spectrum.

BATCODE, UNA BATALLA DENTRO DE TU ORDENADOR (y IV)

Sergio MARTÍNEZ LARA

En este último capítulo de la serie dedicada al Batcode, vamos a abordar el aspecto quizás más interesante y esperado: los ejemplos prácticos. Inspirándonos en ellos podemos llegar a elaborar complejos programas en este nuevo lenguaje.

Hasta aquí se ha ofrecido el repertorio completo de instrucciones y ahora es conveniente mostrar algunos ejemplos de programas. Para empezar puede valer uno de los más pequeños y a la vez más peligroso. Este programa es llamado por Dewdney «Enano» y se puede ver en la *figura 8*.

```
001 SAL   9
... ...  ...
... ...  ...
009 DAT  -1
010 SUM  -1   #10
011 MOV  #0   @-2
012 SAL  -2
```

Figura 8. Programa «Enano».

La primera instrucción es un salto nueve posiciones más adelante, es decir, a la línea diez. Esta instrucción tiene por objeto saltar por encima de la instrucción DAT de la línea 9 y que el programa no quede atascado en ésta.

En la línea diez sumamos el DAT de la posición anterior con 1Ø. Luego, en la instrucción MOV lo que hacemos es lanzar una bomba (#Ø) donde indique el dato de la línea situada dos posiciones atrás (DAT de la posición 1Ø) y, finalmente, repetimos el ciclo saltando otra vez a la línea de suma. Con esto logramos que una barrera de ceros (o DAT's que es lo mismo) barra toda la memoria.

El número de la línea 1Ø es muy importante. Si lo aumentamos el programa «correrá» más lanzando ceros, pero dejará más hueco entre ellos. También es crucial que este número sea múltiplo de cinco. Os dejamos la tarea de averiguar por qué, aunque como ayuda pensar en lo que puede ocurrir si este número fuera, por ejemplo, 4 y tener en cuenta que la memoria es «circular».

Otro ejemplo válido es el programa «Gemini». No se trata de un programa agresivo, ni siquiera se defiende. Es simplemente un programa que se copia a sí mismo un determinado número de posiciones más adelante. Con muy ligeras modificaciones éste puede ser utilizado por nosotros en un momento en el que viéndonos amenazados deseemos trasladar una copia de un programa más grande y complejo a un lugar supuestamente seguro.

Este ejemplo lo podéis ver en la *figura 9*. No empieza en la posición uno, sino en la ocho; en la primera instrucción debéis poner una instrucción de salto igual a la del ejemplo anterior para que la ejecución no se pare en la posición 8 donde se encuentra una declaración de dato. Se ejecuta en primer lugar la instrucción de la línea 1Ø. Esta lo que hace es copiar la declaración de dato de la posición 8. Luego se hace amplio uso de la indirección, pues se copia la línea indicada por el dato —3 (línea 8) donde indique la posición 9. La primera vez que ejecutemos esto habremos copiado la línea 16 (16 = 8 + 8) a la 116 (116 = 9 + 1Ø7). Posteriormente, restamos 1 al dato —3 (es decir, línea 9) y realizamos una operación de «resta y salto si es cero» en la línea 8. Como en primer lugar no será cero, continuaremos con el ciclo hasta que se complete la copia de todas las instrucciones. Luego haremos una actualización del DAT de la línea 1Ø9 con el valor 1Ø7, pues éste contenía un 1ØØ cuando se realizó la copia (probar a quitar esta instrucción y veréis que es cierto. Tratar de averiguar por qué ocurre esto). Finalmente, realizamos un salto a la posición 11Ø donde se empezará de nuevo el ciclo.

```
001 SAL   9
... ...  ...
... ...  ...
008 DAT   8
009 DAT 107
010 MOV  -2    98
011 MOV @-3   @-2
012 RES  -3    #1
013 DSC   2    -5
014 SAL  -3
015 SET  94  #107
016 SAL  94
```

Figura 9. Programa «Gemini».

Si queréis copiar un programa más largo y este ejemplo lo utilizáis como subrutina, deberéis modificar los valores de las líneas 8,9 y el segundo operando de la 15.

Como curiosidad probar el siguiente programa que Dewdney llamó «Trasgo».

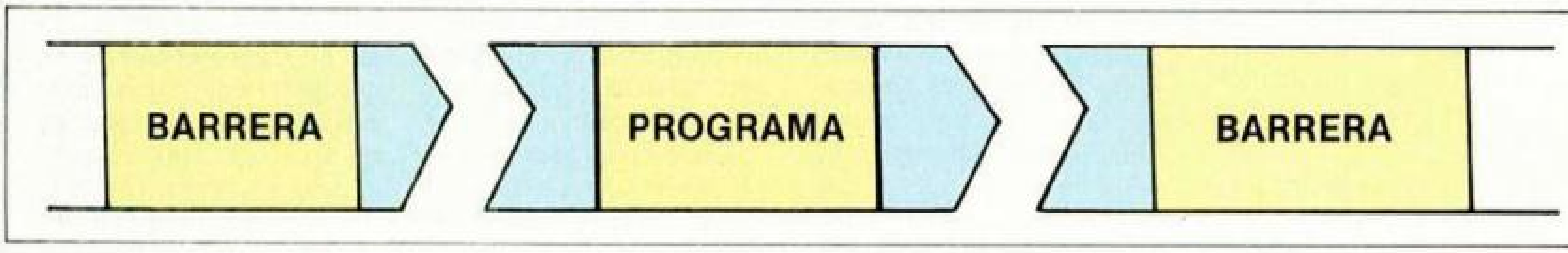


Figura 10. Programa con barreras.

1 MOV Ø 1

Dejarlo correr un tiempo. Detener la ejecución y hacer un «Dump» de memoria. Interesante, ¿verdad? Como experiencia hacer que compita unas cuantas veces este programilla contra el «Enano».

Estrategias más complejas

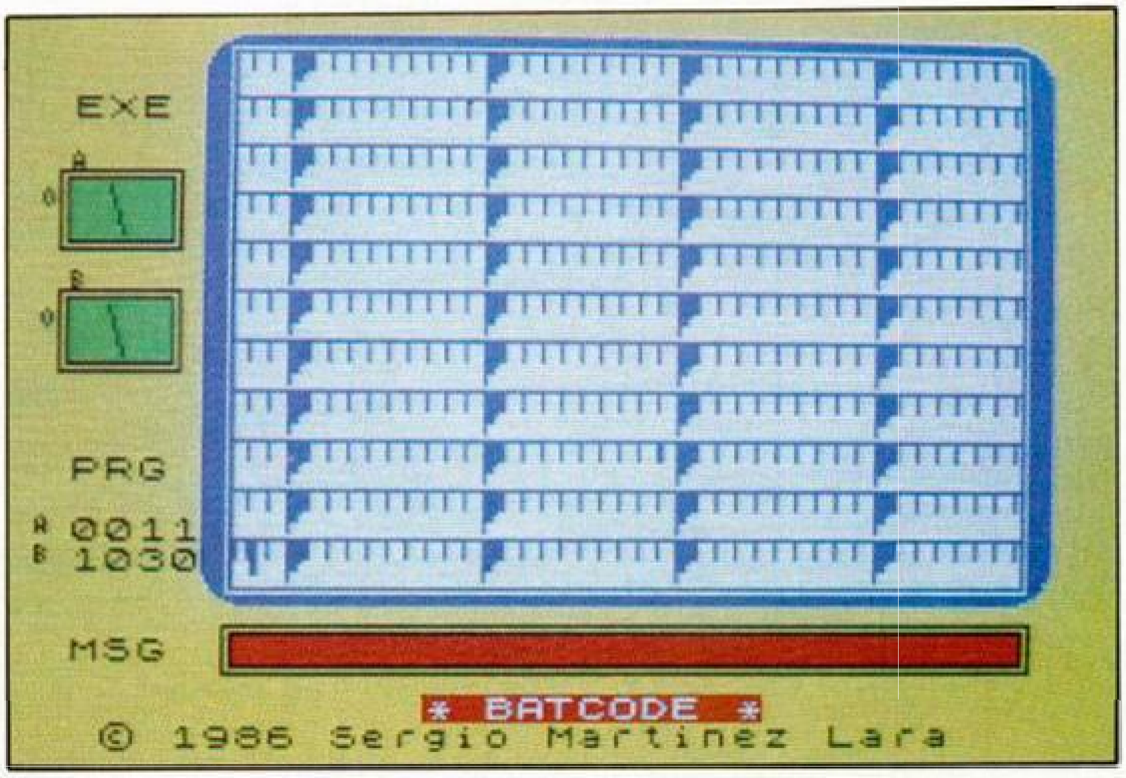
El programa «Gemini» sirve como base para un tipo que puede ser bastante peligroso. Si, por ejemplo, lo modificamos para que se copie así mismo cinco o diez lugares más adelante, es bastante probable que consiga «cazar» al programa enemigo. Probar también en vez de con cien, diez o cinco líneas de intervalo, con un número primo bastante grande.

El problema con este tipo de programas que literalmente «arrollan» al contrario es que estrictamente hablando no vencen al oponente, pues como dijimos al principio, el objetivo del juego es hacer que el programa enemigo deje de ejecutarse y con esto lo que conseguimos es que el oponente ejecute también nuestro código. Es decir, estaríamos en un empate.

Una idea válida puede ser mandar secciones de código que intenten «atrapar» al programa enemigo en un determinado espacio para luego intentar abatirlo bombardeando dicha zona.

Tampoco debemos olvidar nuestra defensa. Pensemos que, por ejemplo, un programa «Enano» lanzando «ceros» cada pocas posiciones puede dar al traste con nuestro programa. Es indespensable tomar algún tipo de medidas encaminadas a evitar este desastre.

Una solución puede ser un programa compuesto por tres secciones, dos de las cuales serían simplemente unas barreras que estén situadas a cierta distancia del programa principal, *figura 10.* Este se encargaría de verificar cada cierto tiempo la integridad de dichas barreras. Si alguna de ellas resulta dañada entonces el programa debe trasladarse en sentido contrario de donde se haya encontrado el daño y reconstruir de nuevo todo el «tinglado». El inconveniente es que sólo es capaz de defenderse de ataques relativamente simples, pero si hemos escogido un buen método de ataque puede ser suficientemente eficaz.

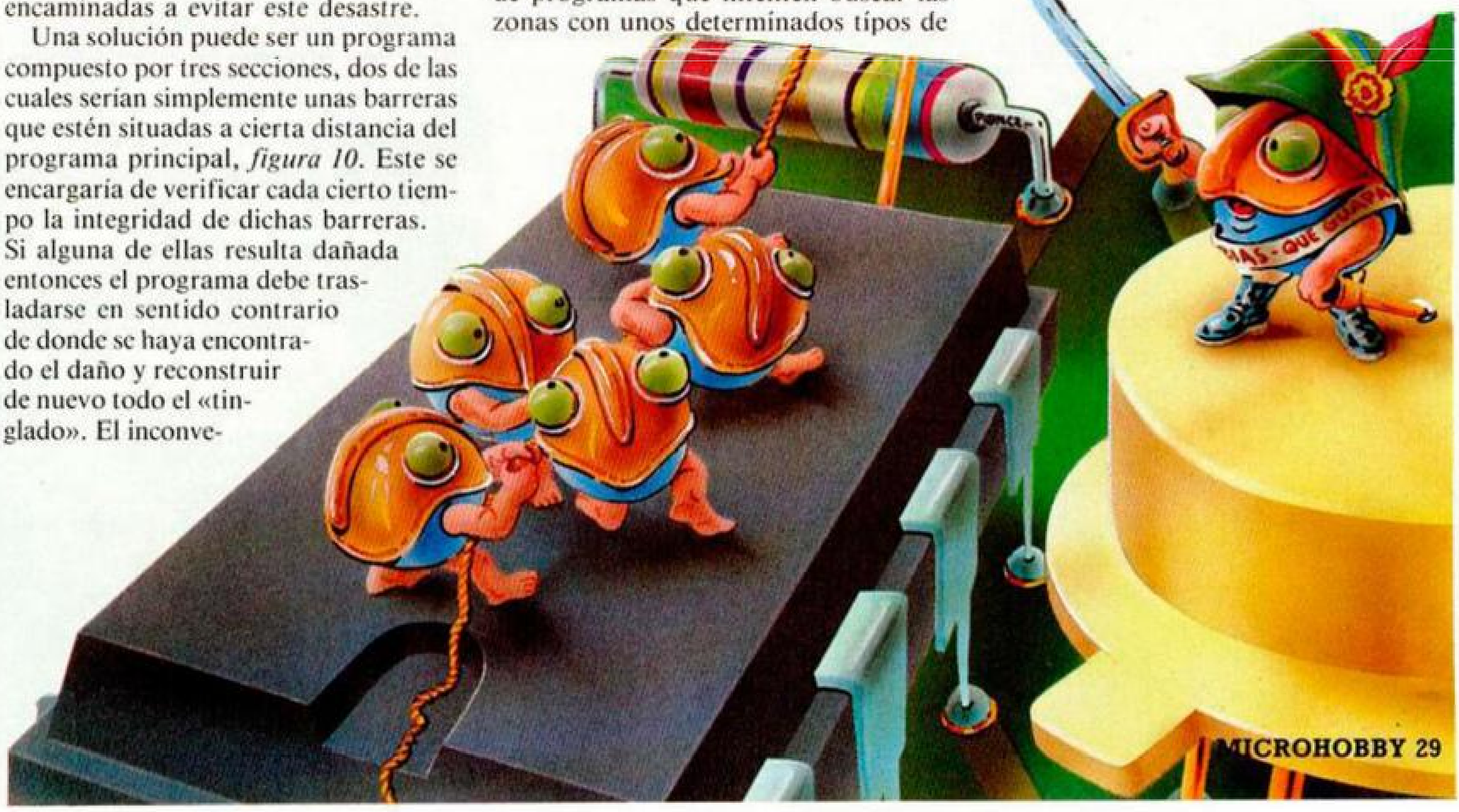


Durante el enfrentamiento entre los programas «Enano» y «Gemini» puede verse claramente el perfil de la batalla: Los bloques más gruesos corresponden al «Gemini» que se autocopia por la memoria, mientras que las barras finas son las bombas lanzadas por el «Enano». El perfil de este último puede apreciarse en la esquina inferior izquierda.

Otro método es el de mantener copias de «seguirdad». Por ejemplo, podemos suponer que el programa actual no está afectado por ningún ataque. Entonces comprobaríamos que la copia está bien y no sufre desperfecto y por último, trasladaríamos la ejecución a dicha copia. O, por ejemplo, tener varias copias y suponer que si dos copias son iguales estarán bien y continuar la ejecución en una de elllas.

No olvidéis tampoco el aspecto ofensivo; se puede ir por ejemplo, a un tipo de programas que intenten buscar las zonas con unos determinados tipos de código e intensificar el bombardeo en ellas. Por ejemplo, esta búsqueda puede concentrarse en las instrucciones de salto.

Todas estas ideas deben ser exploradas por vosotros y por supuesto, debéis encontrar nuevas vías para el desarrollo de este curioso combate. Probar a intecambiar programas con vuestros amigos y compañeros y por favor, no dejéis de comunicarnos vuestros hallazgos a medida que vayáis consiguiendo un completo dominio de este lenguaje, pues es muy posible que nos animemos a convocar una competición a nivel nacional. Así que, iros preparando.



SVI 728



- Capacidad de Memoria de 80 K RAM y 32 K ROM, con el intérprete de BASIC (MICROSOFT) incorporado.
- Teclado de diseño ergonómico, con teclas numéricas independientes.
- Gráficos de alta resolución: red de 296 por 152 puntos, con 16 colores disponibles.
- Conexiones para TV, monitor, unidad de disco de 320 K. (incluido sistema operativo CP/M), impresora, cassettes y cartuchos MSX.
- Conector para dos Joysticks.
- Ordenador MSX.

Aquí y ahora, con toda su potencia

C&M

SVI 738 X'press





- Sistemas operativos: CP/M, MSX-BASIC y MSX-DOS.
- Con una memoria de 80 K RAM.
- Unidad de disco de 3' 5'' integrada en la consola del teclado.
- Dos puertas de conexión: RS232-C y Paralelo Centronics.
- Salidas directas a televisor y monitor.
- Admite directamente la conexión de una segunda unidad de disco, sin necesidad de interface o cartucho de ampliación de memoria.
- Posibilidad de operar en 40 ó 80 columnas sin necesidad de cartucho.
- Transporte cómodo: asa incorporada y maletín de transporte.



SPECTRAVIDEO

Desde ahora SVI-Spectravídeo está aquí.

Con toda la potencia de su nombre y su organización mundial, con toda su tecnología de futuro.

SVI-Spectravídeo marca el comienzo de una nueva era.

Muchas cosas se van a quedar en el pasado.

Y, probablemente, nada será igual a partir de ahora.

Con SVI-Spectravídeo, la vida diaria puede hacerse más sencilla, con más posibilidades, más divertida y excitante, tanto en casa como en el trabajo.

Porque, desde ahora, y para el futuro, puedes contar con SVI-Spectravídeo, España.

Aquí, con toda su potencia.
Y, siempre, muy cerca de ti: en 1.500 puntos de venta.

SVI-Spectravídeo, España, significa la más avanzada tecnología, mantenida por el servicio técnico más exigente y eficaz, en:

Ordenadores. Periféricos y accesorios. Joysticks. Juguetes electrónicos. Sonido Hi-Fi.

SVI S.A.
ESPAÑA

CONSULTORIO

«El Fontanero»

He tecleado el programa «El Fontanero» del número 64 y me da error: «B Integer out of range, 2505:5». He revisado el listado y está como el de la revista. Espero que podáis solucionar el problema y decirme si el error lo he cometido yo o si es de imprenta.

Manuel CHAVERO - Barcelona

☐ Por el error que nos indica, lo más posible es que se haya equivocado al copiar los datos de las líneas 3000 a la 3050. Puede saber exactamente donde está el error si, cuando le salga el mensaje, teclea: PRINT n-65047 que le dará un número. Cuente a partir de la línea 3000 tantos datos como indique ese número y encontrará el dato erróneo. Una de las cosas que pasan con más frecuencia es que se haya olvidado de poner una «coma».

Ojo al «RANDOMIZE 0

Hice el programa «Cargador hexadecimal» de la ficha R-0 e introduje el listado en DATA de la ficha R-23 (sólo el de Scroll a derecha). El resultado es el apetecido, pero introduje el siguiente programa:
*1 PLOT 0,RND*175:RANDOMIZE USR 60000: GO TO 1*

Lo que pasa es que hace el scroll a derecha pero los valores aleatorios no son tales, sino que son valores en un orden fijo. Produciendo, en vez de puntos dispersos desplazándose a la derecha, rayas diagonales.

Daniel PEREZ - Barcelona

☐ Las rutinas de «Scroll» de la microficha R-23 retornan con el registro «BC» a «0», por lo que llamarlas con «RANDOMIZE USR...» es equivalente a hacer un «RANDOMIZE 0» que, como todos saben, transfiere a «SEED» el contenido de los dos octetos inferiores de «FRAMES»; variable, esta última, que se incrementa cada 20 milisegundos.

Normalmente, un «RANDOMIZE 0» suele hacer la secuencia más aleatoria; pero, si se ejecuta sistemáticamente, lo único que conseguiremos es que la secuencia pseudo-aleatoria siga los incrementos del contador de cuadros «FRAMES» que tienen bastante poco de aleatorios. En otras palabras: su problema se resuelve si llama a la rutina con: LET a = USR 60000 en lugar de hacerlo con RANDOMIZE.

Por lo demás, su idea es genial, de hecho, recomendamos a los lectores que prueben el siguiente efecto:

1. Cargar la rutina de scroll a derecha en la dirección 60000 y la de scroll izquierda en 60050.
2. Escribir y ejecutar el siguiente programa en Basic:

```
10 PAPER 1: BORDER 1: INK 7: CLS
20 LET a$ = INKEY$
30 IF a$ = «p» THEN LET a = USR 60000: PLOT 0,RND*175
40 IF a$ = «o» THEN LET a = USR 60050: PLOT 255,RND*175
50 GO TO 20
```

A partir de ahí, podrán moverse por el «firmamento» con las teclas «P» y «O».

Los «displays» de cristal líquido

Quiero pilotar el display de cristal líquido de un reloj de pulsera digital desde el Spectrum. Ya he roto dos relojes (menos mal que costaron 400 ptas. cada uno) y lo más que he conseguido, haciendo combinaciones entre los 13 puntos de conexión que suelen tener y los dos polos de una pila de 3 voltios, ha sido excitar todos los segmentos a la vez o combinaciones incoherentes de ellos. ¿Podríais explicarme cómo excitar cada uno de los segmentos de cada dígito?

José Luis SANZ - Valladolid

☐ De momento, podemos decirle que lo que usted pretende no es tan fácil como parece, por lo que será mejor que tenga bastantes conocimientos de electrónica digital si no quiere arruinarse comprando relojes.

Este tipo de displays van multiplexados, es decir, los dígitos se activan secuencialmente de forma que, cuando uno de los dígitos muestra un número, todos ellos reciben ese número pero sólo se excita el que ha de visualizarlo. La sucesión de dígitos es tan rápida que el ojo cree verlos activados todos a la vez y con distintos números.

La tarea de pilotar un display de este tipo desde un Spectrum requiere bastante circuitería. Primero deberá sacar los datos del Spectrum para lo cual necesitará el correspondiente interface. Luego, necesitará algunos «latch» donde almacenarlos y, finalmente, un multiplexor que sea capaz de excitar displays de cristal líquido. Habitualmente no nos gusta ser pesimistas, pero esta vez, nuestro consejo es que abandone la idea. Resulta demasiado complicado para que valga la pena.

Problemas de carga

Tengo una Spectrum 48K y algunos programas no me cargan. ¿Es a causa de la computadora o de la grabadora? Uno se llama «Saboteur» y otro «Yabba dabba doo».

Otra cosa es que no encuentro por ninguna parte estos cassettes: «Everyone's a Wally» y «Back to skool». Agradecería mucho si me los mandarais.

Sandro BREGANTE - Barcelona

☐ La mayor parte de los problemas de carga son debidos al cassette, no obstante, siempre existe la posibilidad de un defecto de fabricación en la cinta. Si después de probar con otros cassettes las cintas siguen sin cargar, lo mejor será que solicite al distribuidor que se las cambie por otras.

Respecto a su petición, sentimos no poder complacerle ya que no somos distribuidores de software. Con bastante frecuencia, nos escriben lectores pidiéndonos que les mandemos catálogos, cintas, periféricos, etc. Aprovechamos la ocasión para dejar claro que nosotros no podemos vender este tipo de cosas. Para ello deberán dirigirse directamente a nuestros anunciantes o a comercios especializados de micro-informática.

«Clave de acceso»

Les escribo porque desearía que me aclararan una duda referente al juego «DRAGONTORC». Una vez cargado, aparece un mensaje en la pantalla: «Tipe in number at...» Donde están los puntos suspensivos, aparece una letra seguida de una coma y un número. La letra y el número varían cada vez que cargo el juego.

Desearía que me dijeran qué he de hacer y, si se trata de una clave, desearía conocerla.

Carlos Daniel HERNANDEZ - Avila

☐ Efectivamente, se trata de una clave de acceso pa-

ra protegerse de la piratería. Si usted ha comprado el programa en un distribuidor autorizado, le vendrá acompañado de unas instrucciones y una tabla de códigos, ya que éstos varían cada vez que se carga. Si ha comprado el programa por cauces no legales, lo mejor que puede hacer es exigir la devolución de su dinero al «pirata» que se lo haya vendido.

De nuevo el «BASKET»

En el número 68 (sección Consultorio) ponéis un POKE para solucionar el problema del movimiento en el programa «BASQUET» del número 45. El caso es que, aun con el POKE, los jugadores siguen sin moverse.

Nicolás VERDAGUER - Barcelona

☐ Está claro que el mejor escribiente echa un borrón, y en este caso no tenemos más remedio que entonar el «mea culpa» ya que en el Consultorio a que hace referencia, metimos la pata un pelín.

Lo cierto es que no nos dimos cuenta de que el programa hace las lecturas de teclado para mover los jugadores utilizando la función «IN» que, como todos sabemos, devuelve resultados distintos dependiendo de la versión donde corra el programa. Por tanto, los que tengan un Spectrum «ISSUE 3B» deberán restar 64 a todos los datos devueltos por los ports. Concretamente los cambios a realizar afectan a las líneas 510 (cambiar 253 por 189), 520 (247 por 183 y 251 por 187), 540 (251 por 187 y 253 por 189) y 550 (251 por 187 y 247 por 183).

Spectrum 128K

Me he comprado un Spectrum 128K y el aparato presenta una serie de problemas que a continuación les detallo:

1. Al poner «INK 7» los caracteres se visualizan unos en amarillo y otros en azul con el consiguiente embrollo para leer un listado.

2. El sonido es por la televisión pero ocurre que si sintonizo la tele para ver bien la pantalla, el sonido sale fatal y si busco una buena escucha es la pantalla la que se ve fatal.

3. En la parte superior e inferior del borde, aparecen dos franjas de un tono más oscuro que el color que se le haya dado al borde.

Lo llevé donde lo había comprado y me dijeron que todos los 128K son así: ¿Es esto cierto o está el ordenador averiado?

José Manuel RODRIGUEZ - Sevilla

☐ Por supuesto que no es cierto el que todos los 128K sean así. Es evidente que, o su ordenador o el televisor, sufren una avería. Lo más posible es que sea el ordenador; pero, no obstante, conviene probarlo con otro televisor para asegurarse.

Si comprueba que es del ordenador, mándelo a reparar que para eso está en garantía.

DE OCASION

● VENDO Spectrum 48K, con televisor Elbe 12" (comprada en dic-85), incluyo cassette Computone, itnerface tipo Kempston (con reset y piloto de funcionamiento) joystick Quick Shott II, revistas Microhobby y otras, libros. Precio: 45.000 ptas. (negociables). Valorado en 60.000 ptas. Eduardo. Tel.: (94) 247 92 56.

● VENDO interface Impresora centronics / RS 232 Ventamatic, para Spectrum 16/48K o Plus por cambio de ordenador. Regalo cable para conectar a impresora valorado en 3.000 ptas. y programa adaptado a este interface. Todo por 7.000 ptas. Preferentemente dentro de Barcelona. Llamar al Tel: (93) 422 04 66 (22,00 y 23,00 horas).

● VENDO / CAMBIO por televisor b/n pequeño, vídeo-juegos para TV en color con cuatro juegos y dos joystick. Precio: 6.000 ptas. Interesados escribir a la siguiente dirección: Luis Alberto Cano: C/ Doctor Manzanares, 20; 1.° D. Ocaña / Toledo.

● VENDO consola vídeo-juegos Philips G-7000 nueva por 18.000 ptas. También vendo Spectrum 48K apenas usado en 20.000 ptas. con teclado Investrónica (nuevo modelo) en 12.000 ptas. Impresora Spectrum Zx Printer, interface 2 y dos joystick sin usar. Todo en 45.000 ptas. Regalo manuales de uso, libros. Alberto Llorente. Ldo. Poza, 51. 48011 Bilbao.

● ME GUSTARIA ponerme en contacto con usuarios del Zx Spectrum 48K, para intercambiar pokes. Si es posible de Barcelona. Interesados llamar al Tel.: 229 50 28. Preguntar por David.

● VENDO teclado y carcasa de Spectrum 48K, nuevos y en perfecto estado, en 2.000 ptas. o bien lo cambio por los n.° 1, 2, 6 de Microhobby Semanal (completos) y 1.500 ptas. llamar al Tel.: 407 27 00 Ext 521 (oficina). César.

● INTERCAMBIO conocimientos para los siguientes ordenadores: Spectrum 128K —Casio PB 300— Casio PB 700. Llamar al José al Tel.: 332 56 90 o 422 66 41 de Barcelona.

● VENDO Spectrum 48K, lápiz óptico, interface kempston, manuales en castellano, más de 50 revistas. Todo nuevo y precio a convenir. Interesados escribir a Felipe Navarro Soto. Leopoldo Alas, 29. Oviedo 33008 o bien llamar al Tel.: (985) 21 44 63 (horas de comida).

● VENDO Spectrum 48K, teclado profesional Quick Shott V, garantía Investrónica, regalo libros y revistas. También vendo Interface 1 más Microdrive. Escribir a la siguiente dirección: Joaquín Bayón López c/ Capitán Almeida, 28, 1°B. Oviedo.

● INTERESARIA intercambiar información, sobre las últimas novedades, ofertas y adelantos en tecnología en España. Escribir a Gullermo Echartea Martinelli.

● VENDO Sinclair QL, completo adquirido en Sep-85, más programa Chess, 3 libros sobre utilización comercial y 14 microdrives. Todo por 60.000 ptas. Preguntar por Carlos. Tel.: (91) 463 15 36 sólo tardes.

● VENDO Spectrum 48K en buen estado. Cables, alimentación y manuales. Regalo revistas. Precio: 20.000 ptas. (negociables). Interesados escribir a Pablo Pérez de Ayala González. c/ Martínez Campos, 24, 3°D. Granada 18002. Tel.: 25 72 75.

● ¿QUIERES aprender alemán o inglés con tu ordenador? Tengo un curso que te puede interesar. Escribir para enviar más información totalmente gratis. Carmen Abella. Portugal, 32, 4°D. Móstoles / Madrid.

● VENDO enciclopedia práctica de Electrónica por 7.000 ptas, nueva, encuadernada. También vendo los 45 primeros números de «Mi Computer» por 5.500 ptas. y los 21 primeros de la «Enciclopedia práctica de la Informática «por 2.000 ptas. Interesados escribir a Ramón piedrefita c/ Paseo de Franco, 10. 3°. Jaca/ Huesca. Tel.: (974) 36 03 14.

● VENDO para Zx Spectrum la siguiente lista: lápiz óptico de Investrónica por 4.500 ptas. Interface II, pro 2.200 ptas. Zx Microdrive con libro de instrucciones en inglés. Interesados escribir a la siguiente dirección: Manuel Muñoz Venzalá c/ Peña Atalaya, 101, 3°B. Madrid.

● VENDO lápiz óptico completamente nuevo con manual, interface. Precio a convenir. También estoy interesado en adquirir cajas para guardar cintas. Interesados llamar de 2 a 3 al Tel.: (974) 60 44 91. Preguntar por José Luis.

● VENDO Zx Spectrum 48K, incluyo revistas, libros, interface para joystick tipo Kempston y otro programable, reset, cassette especial para ordenadores marca Gold King, 4 libros de programación y 14 microdrives. Todo por 50.000 ptas. Preguntar por Carlos al Tel.: (91) 463 15 36.

● VENDO rutinas de Código Máquina y Basic. Pueden solucionar muchos problemas. También encargos. Precio a convenir. Interesados llamar al Tel.: (93) 347 63 84 o escribir a Victoriano Sancho. C3 Castillejos, 361, 3.° 3ª. 08025 Barcelona.

● SE VENDE ordenador Spectrum 48K más unidad de disco Opus discovery 1, comprada hace un mes con garantía, josytick Quick Shot II, 4 discos vírgenes, 50 revistas de Microhobby y todo en buen estado con embalaje y manuales en español por sólo 75.000 ptas. Interesados llamar al Tel.: (91) 276 57 01.

● ME GUSTARIA intercambiar información, últimas novedades, ofertas y adelantos en tecnología en España. Interesados escribir a la siguiente dirección: Guillermo EChartea Martinelli c/ Battle, 325. Trinidad. Dpto. de Flores / Uruguay. Tel.: 2783.

● VENDO Spectrum 48K, nuevo, con todos sus accesorios por 19.000 ptas. También Interface programable de sonido y joystick. Interesados llamar al Tel.: (968) 26 38 39 de Murcia.

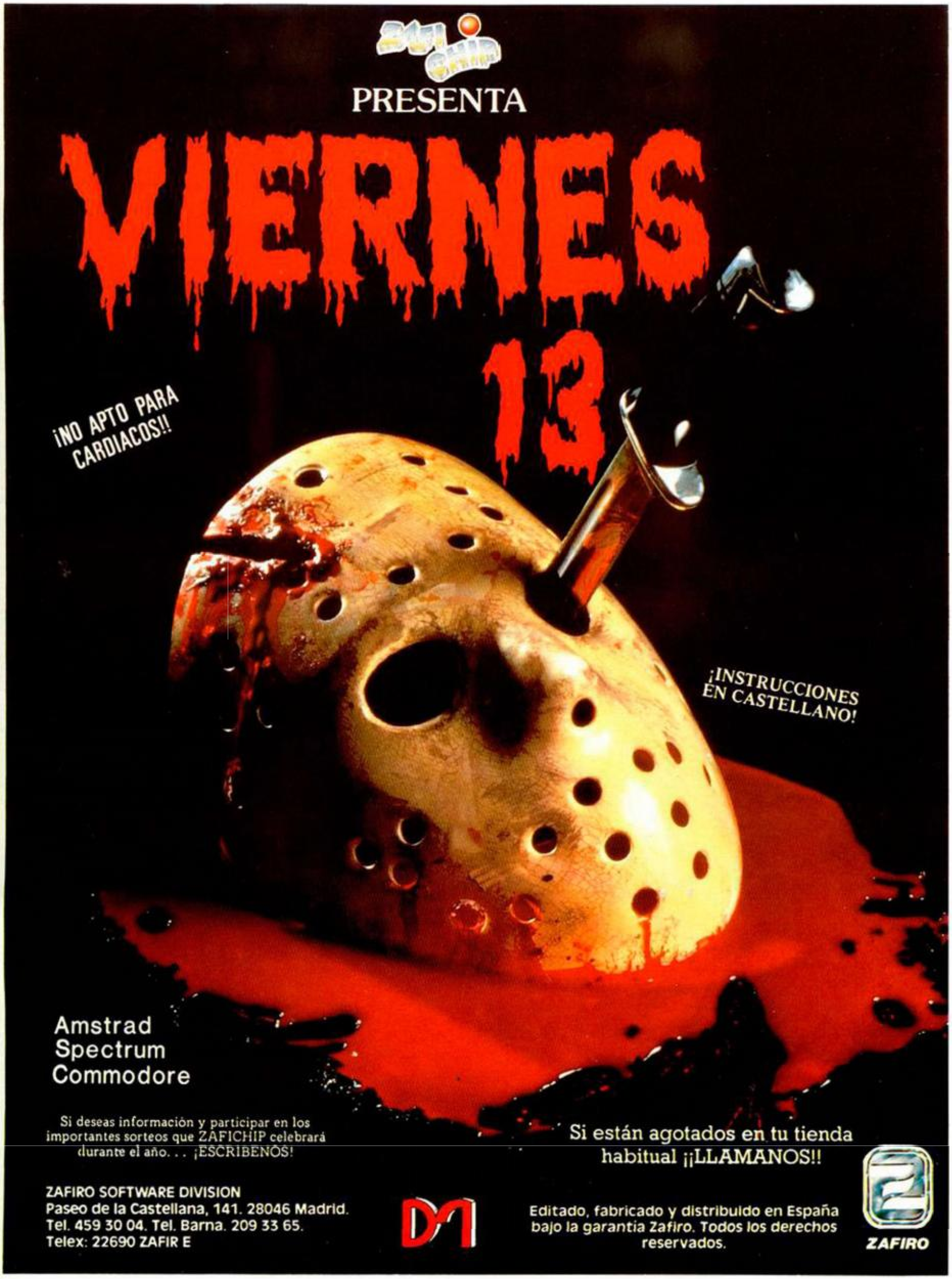
● CAMBIO Spectrum Plus, cassette e interface josytick por Commodore 64K con cassette o por unidad de diskettes para este ordenador. También lo vendo por unas 50.000 ptas. Jorge Rom. Tel.: (93) 204 14 51.

● VENDO Atari vídeo Computer System, en perfecto estado, completo, cables, fuente de alimentación y dos tipos de mandos. Precio: 12.000 ptas. Interesados llamar o escribir a Carlos Cabestany c/ De los Caballeros, 88, 5.°, 2.°. Barcelona Tel.: (93) 204 06 96.

● VENDO Commodore 64, datasette, joystick Quick Shott II, libros, revistas, por sólo 48.000 ptas. También vendo Interface para adaptar. TV. CBM/64 y 128K o VIC 20 a cualquier cassette, por 4.000 ptas. Interesados escribir a la siguiente dirección: A. Cañete. Avda. Isabel la Católica, 108, Sob 1.° Hospitalet / Barcelona. Tel.: (93) 334 07 70.

● VENDO Vic-20, ampliación 16K, tarjeta de carga rápida, libros: «Guía del Usuario», «Guía de Referencia del programador», «Curso de Basic» en dos tomos. Llamar por las noches al Tel.: 254 80 62 de Madrid. Preguntar por Rafael Rojo.

MH


ZAFI CHIP
PRESENTA
VIERNES 13
¡NO APTO PARA CARDIACOS!!
¡INSTRUCCIONES EN CASTELLANO!
Amstrad
Spectrum
Commodore
Si deseas información y participar en los importantes sorteos que ZAFICHIP celebrará durante el año. . . ¡ESCRIBENOS!
Si están agotados en tu tienda habitual ¡¡LLAMANOS!!
ZAFIRO SOFTWARE DIVISION
Paseo de la Castellana, 141. 28046 Madrid.
Tel. 459 30 04. Tel. Barna. 209 33 65.
Telex: 22690 ZAFIR E
Editado, fabricado y distribuido en España bajo la garantia Zafiro. Todos los derechos reservados.
ZAFIRO



SPECTRUM 128
EL SUMMUM

Spectrum, como líder, marca un nuevo hito en la historia de los ordenadores familiares.

El Spectrum 128.

Gran capacidad de memoria. Teclado y mensajes en castellano, teclado independiente para operaciones numéricas y de tratamiento de textos...

Sinclair e Investrónica han desarrollado una auténtica novedad. En ningún lugar del mundo, salvo en los Distribuidores Exclusivos de Investrónica, podrás encontrar el nuevo Spectrum 128.

Sé el primero en tener lo último.

SPECTRUM 128. NOVISSIMUS

DISTRIBUIDOR EXCLUSIVO

investronica

Tomás Bretón, 62.
Tel. (91) 467 82 10.
Telex 23399 IYCO E.
28045 Madrid

Camp, 80.
Tels. (93) 211 26 58 - 211 27 54.
08022 Barcelona